A Study on Multi-Subspace Representation of Nonlinear Mixture with Application in Blind Source Separation: Modeling and Performance Analysis

by

Lu Wang

Dissertation

Submitted by Lu Wang

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Engineering (Ph.D.)

Supervisor: Prof. Tomoaki Ohtsuki, Ph.D.

Graduate School of Science and Technology
Keio University

August, 2019

Contents

List of Tables

List of Figures

A Study on Multi-Subspace Representation of Nonlinear Mixture with Application in Blind Source Separation: Modeling and Performance Analysis

Lu Wang
wanglu@keio.jp.
Keio University, 2019

Supervisor: Prof. Tomoaki Ohtsuki, Ph.D.
ohtsuki@ics.keio.ac.jp

Abstract

Recognizing multiple signals from the multiple observations or mixtures received by a set of sensors is the task of source separation. The problem is referred to as "blind" source separation when the procedure has access only to the observations without any prior knowledge information for the mixing system. The basic idea of nonlinear blind source separation (BSS) is to generalize the highly successful linear independent component analysis (ICA) framework to arbitrary, but usually smooth and invertible, nonlinear mixing functions. Thus, the observed data is assumed to be a nonlinear invertible transformation of statistically independent latent quantities, and the goal is to find the mixing function, or its inverse, solely based on the assumption of the statistical independence of the latent quantities.

However, nonlinear BSS is one of the biggest unsolved problems in unsupervised learning. The statistical independence of estimated sources is no longer a sufficient constraint for demixing functions. In fact, there is an infinite number of possible nonlinear decompositions of a random vector into independent components, and those decompositions are not similar to each other in any trivial way. The aim of this thesis is to develop the practical methods of modeling and performance analysis for nonlinear BSS. Some of the work consists of incremental extensions to existing linear methods. The improvements are formulated in general terms in order to be useful in other kinds of learning problem as well.

Chapter 2 provides an overview of existing algorithms for BSS. Some mathematical preliminaries are introduced for BSS of nonlinear mixing models. Special emphasis is to a multi-subspace mapping approach that applies ensemble learning to a flexible multilayer perceptron model for finding the sources and nonlinear mixing mapping that have most probably given rise to the observed mixed data.

The approach in Chapter 3, is inspired by the idea of an efficient multi-subspace representation to approximate the nonlinearity or distortion caused by mixing function. Relying on the multi-subspaces architecture, the algorithm transforms a time-invariant nonlinear BSS to the local linear problem with a tolerable computational cost. Then the projected data can break the nonlinear problem down into the version of a generalized joint diagonalization problem in the feature space.

Importantly, the parameters and forms of polynomials depend solely on the input data, which guarantee the robustness of the structure. We thus address the general problem without being restricted to any specific mixture or parametric model.

In practice, the approximation function is derived from some estimation algorithm with a finite sample size that even larger estimation error appears with improper model construction. In Chapter 4, we work on the convergence and asymptotic analysis of the proposed separation approach in Chapter 2, where the nonlinearity of the mixture function is extracted by the flexible approximation and the nonlinear problem is solved linearly in the parameter space. The analysis stems from the performance of a mismatched estimator that accesses the finite sample size. By providing a closed-form expression of the mean squared error (MSE), we can present a novel algebraic formalization as well as derive an upper bound on the estimation error. The simulation results show that if the nonlinearity of mixing functions can be extracted by the flexible approximation, the consistency of numerical MSE and analytical MSE can be achieved as the sample size tends to be infinity. This implies that the algorithm is feasible to separate the distortion of the nonlinear mixture.

In general, most BSS algorithms assume that the number of sources is less than that of sensors, denoted as overdetermined BSS. However, in practice, this assumption is difficult to be satisfied since the number of sources is unknown. In Chapter 5, we propose a model that relies on a Kernelized multi-subspace and sparse representation in the time-frequency (TF) domain to solve the underdetermined BSS problem. By parameterizing multi-subspaces, we can map the observed signals in the feature space with the coefficient matrix from the parameter space. We then exploit the linear mixture in the feature space that corresponds to the nonlinear mixture in the input space. Once such subspaces are built, the coefficient matrix can be constructed by solving an optimization problem on the coding coefficient vector. Relying on TF representation, the target matrix can be constructed in a sparse mixture of TF vectors with the fewer computational cost. The experiments are designed on the observations that are generated from an underdetermined mixture, and that is collected with some direction angles in a virtual room environment. The proposed approach exhibits a higher separation accuracy.

Another model working on underdetermined BSS problem is introduced in Chapter 6, which is inspired by the idea from a deep architecture. By constructing an ϵ-vanishing polynomial networks (ϵ-VPNs), we can extend the linear BSS method to the nonlinear case. The approach use a set of approximated base to obtain the values attained by mapping functions. Then, we construct the architecture with increasing expressiveness, where the layer of our network begins with the polynomial of degree 1, up to build an output layer that can represent data with a small bias by a good approximate basis. Relying on several transformations of the input data, with higher-level representation from lower-level ones, the networks are to fulfill a mapping implicitly to the high-dimensional space. Once the ϵ-VPNs are built, we can fulfill a simple linear separation algorithm on top of this output as back propagation.

Finally, Chapter 7 summarizes the conclusions and possible perspectives for future research of this work.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor, Professor Tomoaki Ohtsuki, who gives me the opportunity to carry out my doctoral research and allow me to grow as a researcher. His guidance provides the continued support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. Every paper I published, there is no part that does not contain his careful revision. Writing of this dissertation thesis also condenses his comments and suggestions on almost every page. His guidance helped me in all the time of research and writing of this thesis. I have to confess that his interest in research is motivation, and his attention to all students is an encouragement.

I would also like to extend my gratitude to the rest of my thesis committee member, Professor Iwao Sasase, Professor Masaaki Ikehara, and Professor Panagiotis Takis Mathiopoulos, for their insightful comments, which help me improve the thesis to a better scientific level. Thanks for your brilliant suggestions and encouragement, which not only letting my defense be an enjoyable moment, but also incensing me to widen my research from various perspectives.

My special thanks to the Department for Education for their financial support. I am honored to receive the support from JASSO Honors Scholarship, and Research Encouragement Scholarship during the first year of my study, which was subsequently extended to the second year. For the last year of my Ph.D. study, it is an honor to be admitted to Fujiwara Scholarship Foundation. These scholarships are like a confidence booster for sure.

Assistance provided by the following organizations is greatly appreciated, such as Keio Leading Edge Laboratory (KLL) and Telecommunications Advancement Foundation, for offering the research grant to support my trip to IEEE Globecom 2018, APSIPA 2018, IEEE ICASSP 2019, IEEE ICC 2019, and another 10 domestic conferences. The research grant helps me to share the work we have done and discuss with some researchers in the related field. International and domestic conferences are an excellent platform to present our results and gain new insight into possible research interests.

I am grateful to my friends for their invaluable help and relief during tea breaks, and for all the fun we have in the last two more years. A special thanks to my family. Words cannot express how grateful I am to my parents for all supporting spiritually. Thanks for inspiring me to strive towards my goal. I love you forever!

Lu Wang

Notations and Conventions

a, b, T	Scalar or constant
$\mathbf{s}$, $\mathbf{x}$	Column vector
$\mathcal{D}_{\mathbf{s}}$	Column vector in the TF domain
$\mathbf{A}$, $\mathbf{W}$	Matrix
$\mathbf{A}^{-1}$	Matrix inverse
$\mathbf{A}^{\top}$	Matrix transpose
$\mathbf{A}^{\dagger}$	Matrix Moore-Penrose pseudoinverse
$\mathrm{diag}(\mathbf{a})$	A diagonal matrix with the elements of $\mathbf{a}$
$\mathrm{tr}\{\cdot\}$	Trace
$\det(\mathbf{A})$	Determinant of matrix $\mathbf{A}$
$\dfrac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\mathbb{E}[\cdot]$	Expectation
$\bar{\mathbf{\Sigma}}_{\mathbf{s}}$	$\frac{1}{T}\sum_{t=1}^{T}\mathbf{s}(t)\mathbf{s}(t)^{\top}$ empirical counterpart of expectation
$\mathrm{Cov}[\mathbf{a},\mathbf{b}]$	$\mathbb{E}[\mathbf{a}\mathbf{b}^{\top}]$ for any stochastic vector $\mathbf{a}$, $\mathbf{b}$
$\mathbf{1}_N$	Vector one of N elements
$\|\cdot\|$	L^2 norm of the vector
$\|\cdot\|_F$	Frobenius norm
$\mathcal{F}$, $\mathcal{G}$	Nonlinear generative mappings
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^N$	Set of N vectors
$\mathbb{R}^{N\times M}$	Set of $N \times M$ matrices
$\mathcal{N}(\mu,\sigma^2)$	Normal or Gaussian distribution with parameters μ and σ^2
$\mathcal{U}[a,b]$	Uniform distribution with parameters a and b
$\mathcal{KL}(\mathbf{A}\|\mathbf{B})$	Kullback-Leibler divergence of matrix $\mathbf{A}$ and $\mathbf{B}$

Chapter 1

Introduction

1.1 Overview of Blind Source Separation

Blind Source Separation (BSS) and independent component analysis (ICA) are techniques to extract and recover the underlying source signals from multivariable statistical data. Here, "Blind[1]" implies that the problem consists in retrieving unobserved independent mixed signals from mixtures of them, assuming there is information neither about the original source signals, nor about the mixing system. In scientific and engineering applications, many of the observed signals can be seen as a mixture of a plurality of source signals. The observed mixed signal is a series of sensor outputs, with each sensor receiving different combinations of the source signals. The main task of BSS is to recover the source signal that we are interested in from the observed data.

The problem of BSS has been introduced for linear instantaneous mixtures. Many researchers have been attracted by the subject, and many other works appeared. For example, see [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13], which are some of the most important papers on linear instantaneous BSS. [14] is a published book on the subject. A good overview of the problem can be found in [15]. The early works on the BSS and ICA problems concerned linear instantaneous mixtures, and by now, some algorithms are available for separating them. As an extension to the instantaneous mixtures, the convolutive mixtures have been considered by some researchers [16, 17, 18, 19].

A typical example is the "cocktail party" problem. Assume that you are attending a cocktail party with a variety of sounds coming from the surroundings, talking, music, and even a whistle from outside the window. If sufficient microphones are placed at different positions to record these sounds, then each microphone can record signals mixed according to different weights in Fig. 1.1. Although there may be a great deal of interference, you would be able to focus on the words of your friend. Given only the received speech signal from the microphone, how is it possible to separate the desired speaker's voice when the locations of the microphones and the sound source with the information we require are not known beforehand? BSS is introduced to solve this exact problem, that can be formulated as below.

Consider the following linear instantaneous mixing system with m inputs and n outputs as

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \quad t = 1, 2, \cdots, T, \tag{1.1}$$

where $\mathbf{s}(t) = [s_1(t), s_2(t), \cdots, s_m(t)]^\top$ are the signals with m channels, $s_i(t)$ denotes the sample of the i-th source at time index t. The superscript $[\cdot]^\top$ denotes the transpose operation. $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^\top$ denotes the observed mixtures with n channels, which is assumed to be generated by an $n \times m$ mixing matrix $\mathbf{A}$ and the source signals $\mathbf{s}(t)$.

Commonly, the separation process of BSS is conducted on the assumption that the sources vectors are statistically independent [20]. For a linear mixing model, if the number of sources equals that of channels ($m = n$), the demixng matrix $\mathbf{W}$ can be defined as $\mathbf{W} = \mathbf{A}^{-1}$. The recovered signals are represented as $\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t)$. The linear BSS aims at estimating $\mathbf{W}$ and the recovered signals $\hat{\mathbf{s}}(t)$ using only the observed signals $\mathbf{x}(t)$.

Nonlinear phenomena are encountered in many engineering problems. Traditional signal processing techniques are linear, which makes them unable to extract the complex, nonlinear patterns

[1]Strictly speaking, a totally blind solution is not possible, because we need some assumptions about the general form of the mixing system (linear instantaneous, convolutive,$\cdots$).

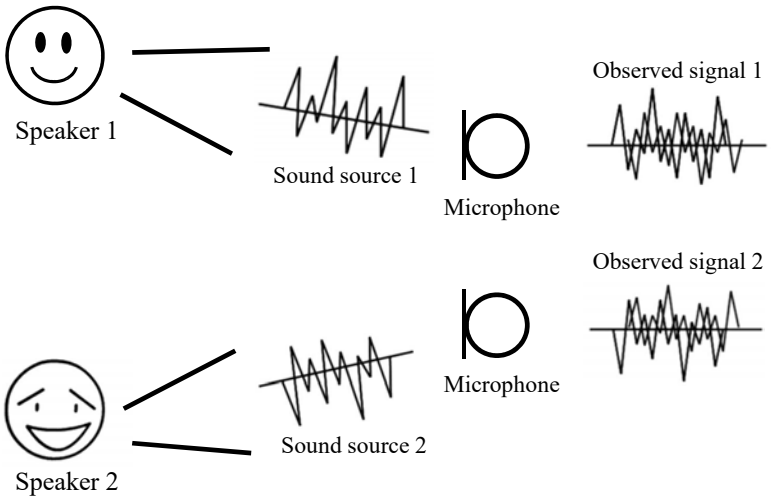


Figure 1.1: The Diagram of Signal Mixing.

that may lie in the data available in such scenarios. Therefore, problems concerning nonlinear data analysis have traditionally been tackled by polynomial filters [21], which provide straightforward extensions of many linear methods, or by neural network approaches [22], which are able to learn nonlinear relationships.

An obvious extension for the task of BSS is that the observed signals are assumed to be generated from a set of sources by a nonlinear, instantaneous and invertible function $\mathcal{F}$, i.e., $\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t))$ for all $t = 1, \cdots, T$. Roughly, the nonlinear BSS seeks to find the mixing function (or its inverse function $\mathcal{G} = \mathcal{F}^{-1}$), solely based on the assumption that the sources are statistically independent. The performing is to design a separation system $\mathcal{G}$, so as to obtain the independent recovered source vector $\hat{\mathbf{s}}(t) = \mathcal{G}(\mathbf{x}(t))$. In other words, each component of the output vector $\hat{\mathbf{s}}(t)$ must depend on only one component of $\mathbf{s}(t)$. In general case, this relation may be nonlinear or have memory.

However, the indeterminacies imposed by the nonlinear model are difficult to handle [23, 24]. Although nonlinear mixtures have been considered in some literatures, [11, 25, 26, 27, 28], the availability results obtained so far are few. One reason is of course the mathematical difficulty of nonlinear systems, but another obstacle for the nonlinear BSS problem is that solutions are non-unique without extra constraints [29]. Since the sole information about the sources is their statistical independence, one can try to construct the separation system $\mathcal{G}$ in such a way that the output vector has independent components. Some suggestion for recovery inconsistency is to add the further prior information directly to the model or as a regularization term in the optimization processing procedure. However, it leads to another problem that the objective functions tend to be heuristic. It is clear that the identifiability of the mixing models is hardly given any theoretical justification or proof.

In addtion, some nonlinear algorithms utilize single approximation to extract the nonlinearity, such as multi-layer perceptron (MLP) in neural network [27, 30], which is employed for estimating the nonlinear separation function. By restricting the smoothness of the target transforming, MLP provides the regularized solutions to ensure that nonlinear ICA leads to the sources separable. However, the example presented in [31] shows that the smoothness property is not a sufficient

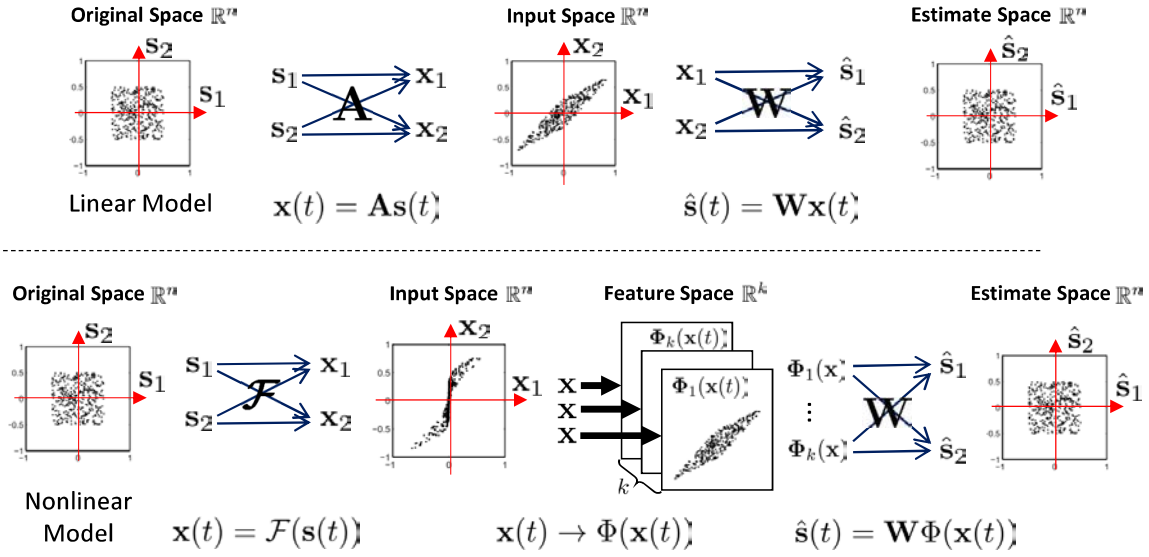


Figure 1.2: The Diagram of Signal Mixing: Linear Model and Nonlinear Model.

condition for this purpose. Hyvärinen and Pajunen [29] show a conformal mapping may be helpful. Nonlinear ICA [32] is able to estimate a separation mapping when the mapping functions are restricted to the set of conformal mapping. Unfortunately, the angle preservation conditions seem very restrictive [33]. In particular, it is not realistic in the framework of the nonlinear mappings associated with the sensor array.

To solve this problem, we propose a novel model with multi-layer architecture to extend the linear BSS. The key idea is to exploit nonlinear mapping proposed in this thesis to extract the nonlinearity of mixing functions, and then the nonlinear problem can be linearly separable. To introduce the key idea, let us consider an example. In Fig. 1.2, the data are assumed to be generated from a uniform distribution. There are two dimensions, $\mathbf{s}_1$ and $\mathbf{s}_2$. The original sources are transformed into the observations using the simplest model, in which the N observed signals $x_1(t), \cdots, x_N(t)$ are assumed to be linear instantaneous mixtures of N zero-mean and statistically independent source signals $s_1(t), \cdots, s_N(t)$. Therefore, the scatter plot exhibits a square shape in the center. The linear mixing like that the signals make the operations of plus or minus with some constants. It likes rotating the data. We can achieve the separation by finding a correct direction and then rotate it back. However, for the nonlinear mixing the solutions are not unique. Therefore, a feasible way perhaps need to resort a flexible approximate. It would be like finding some subspaces they are spanned by some polynomials. If one of the subspaces can match the nonlinearity of mixing function, then the projected data can make the problem linearly separable.

The nonlinear model in Fig. 1.2 shows an intuitive example. Since the observations are nonlinearly mixed in the input space, we generate some approximation functions to extract the nonlinear characteristics in the manifold $\mathcal{H}$. Here, vanishing components allow us to construct the nonlinear variants by some polynomials, such as $\Phi_1(\mathbf{x}(t)), \cdots, \Phi_k(\mathbf{x}(t)) \in \mathcal{H}$, *i.e.*, the data $\mathbf{x}(t)$ are mapped implicitly into the feature space. The feature space is spanned from such polynomials that enable us to work on $\mathcal{H}$. Then BSS approaches can be applied to the projected data in the feature space, which corresponds to the nonlinear BSS approaches in the input space.

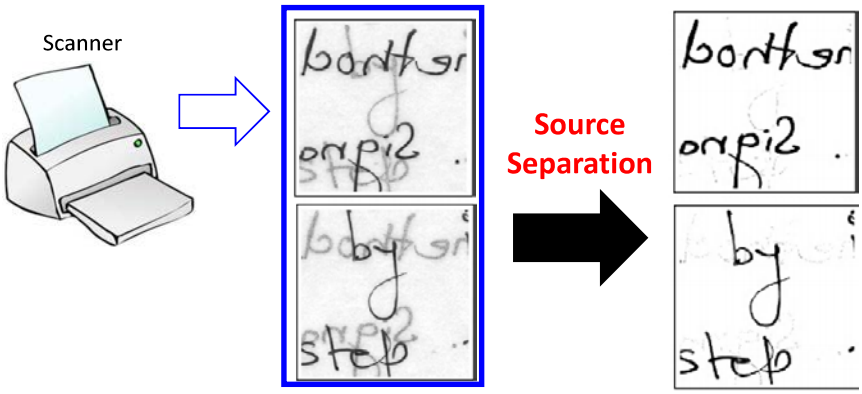


Figure 1.3: Left column is two images distorted by show-through. Right column is results of removing show-through effect in scanned papers.

1.2 Nonlinear BSS and Its Applications

In many applications the mixing system of the sources has to be modeled as nonlinear. Hyperspectral imaging [34, 35], remote sensing data [36], determining the concentration of different ions in a combination via smart chemical sensor arrays [37], and removing show-through in scanned documents [38] are some well-studied examples of such applications.

1.2.1 Removing Show-Through in Scanned Documents

One of applications is on removing show-through in the scanned documents. If we consider a sheet of paper, of which both sides have been printed. If the paper sheet is thin, each side is in fact a mixture of the front and back images.

As we can see in Fig. 1.3, show-through appears when a fraction of the verso is mixed with the recto pixel by pixel in the scanning process. However, this fraction is proportional to the grayscale of the front image, i.e. as the front image becomes darker, the show-through will be lower. Therefore, the mixing model cannot be linear [38].

1.2.2 Nonlinearities in Speech Production

In absence of sounds, the vocal tract could be modeled as a single tube and the air molecules in it can be thought as a linear oscillator, which responds to a disturbance with small displacements from the rest position [39, 40]. The conditions are extremely more complex when speech sounds are produced, sine the motion of the vocal organs changes the vocal tract shape. A coarse model of the vocal tract in these conditions is a set of overlapping tubes of different lengths and cross-sectional areas. Due to its length and section area, each tube is subject to a different air pressure, which in turn, generates different forces acting on the air molecules, causing their very complicate motion. In this case, a linear description fails to describe this complex dynamics and a nonlinear approach should be used.

In addition, nonlinearities in the speech signal are present at least in two descriptive domains: the phonetic and the supra-segmental domain. In the phonetic realm, the rapid dynamic of transient

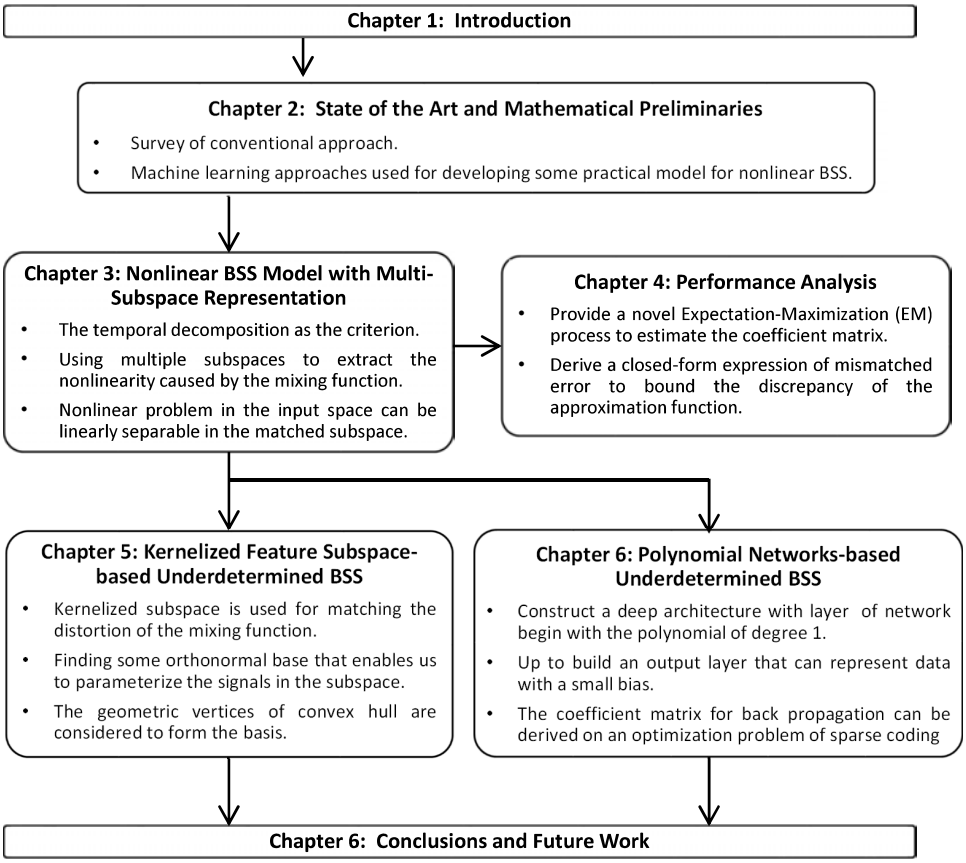


Figure 1.4: The Configuration of this Dissertation.

and turbulent sources (as in the production of consonants), the rapid variation of formant frequency values (usually observed in vowels because of the adjacent phonetic context), confusable sounds, and segmental phenomena such as elision, assimilation, etc., can not be accounted by the stationary and linear assumption.

In the supra-segmental realm, rapid variations in speaking rate, speech energy, and fundamental frequency value as well as variability in the acoustic realization of utterances from different speakers, homophone words, and the intrinsic nonlinearities of the speech production system can not be encoded through a processing algorithm that can only extract linear features. To this aim, several nonlinear speech processing techniques have been proposed, such as nonlinear parametric and no-parametric autoregressive models, speech fluid dynamics, modulation, and fractal methods.

1.3 Scope and Contributions of the Dissertation

1.3.1 Summary of Dissertation

The aim of this thesis has been to develop the practical methods of modeling and performance analysis for nonlinear BSS. Some of the work consists of incremental extensions to existing linear methods. The improvements are formulated in general terms in order to be useful in other kinds of learning problems as well. The outline of this dissertation is summarized in Fig. 1.4.

Table 1.1: Summary of Chapter 3.

Objective	A novel separation model is presented that relies on the temporal structure and a novel mathematical construction with a multi-layer architecture.
Conventional Approaches	1. Ziehe et al. [41]: Temporal Decorrelation source SEParation (TDSEP) is one of the earliest frameworks based on temporal structures. • The temporal decomposition as the separation criterion. 2. Harmeling et al. [42, 43, 44]: a kernelized TDSEP (KTDSEP) method was proposed for nonlinear blind source separation that the kernel functions are used for mapping the observations into the kernel spaces.
Conventional Approaches Limitations	1. The mapping function do not have any optimizing property in terms of the contrast function that allows them to be ranked and evaluated [45]. 2. The method assumes the number of kernel spaces is chosen enough to approximate the nonlinearity without technical reasons.
Proposed Approach	1. A novel mathematical construction with a multi-layer architecture to extract the nonlinearity without extra constraint. 2. The approach processes the data using a flexible approximation that projects the data into some high dimensional feature spaces. 3. Using multiple subspaces, nonlinear problem in the input space can be linearly separable in the feature space.
Improvements and Discussion	1. The proposed algorithm has a higher estimation accuracy on various nonlinear mixtures. 2. Due to adopting the nonlinear approximation in the form of a sample representation, the complexity and storage requirements of the model are proportional to the number of mapping functions. 3. The method may fail if some sources lack specific time structure [46].

Table 1.2: Summary of Chapter 4.

Objective	A novel algebraic formalization is presented as well as derive an upper bound on the estimation error.
Conventional Approaches	1. Fisher information matrix (FIM) [47]. 2. Cramér-Rao lower bounds (CRLB) [48] on the estimation error of the mixing matrix. 3. The correct separation point is discussed theoretically [49].
Conventional Approaches Limitations	1. In some works, the performance of the algorithms is examined numerically [50]. 2. There are no complete performance analysis in the sense of closed-form expressions for an expected figure of merit or bound [51].
Proposed Approach	1. Extension of Expectation-Maximization (EM) process is proposed to estimate the coefficient matrix. 2. A novel closed-form expression of mismatched error to bound discrepancy.
Improvements and Discussion	1. Numerical simulations demonstrate that bound is achievable when the model assumptions hold, as expected from our theoretical analysis. 2. The performance analysis only consider the discrepancy from such polynomials, but not the discrepancy caused by the coefficient matrix.

- In Chapter 3, we present a novel approach to tackle the ill-posed of the nonlinear BSS problem with a few assumptions. The derivation of our algorithm is inspired by the idea of an efficient multi-subspace representation to approximate the nonlinearity or distortion caused by mixing function. Relying on the multi-layer architecture, the algorithm transforms a time-invariant nonlinear BSS to the local linear problem with a tolerable computational cost. Then the projected data can break the nonlinear problem down into the version of a generalized joint diagonalization problem in the feature space. Importantly, the parameters and forms of polynomials depend solely on the input data, which guarantees the robustness of the structure. We thus address the general problem without being restricted to any specific mixture or parametric model.

- In Chapter 4, the process deals with BSS in the nonlinear mixture is to estimate the original signals or mixture functions from the degraded signals, without any prior information about the mixing functions. The fundamental problem is to recover the original sources by estimating an approximation function under such assumptions so as to estimate the inverse of mixing functions. However, in practice, the approximation function is derived from some estimation algorithm with a finite sample size that even larger estimation error appears with

Table 1.3: Summary of Chapter 5.

Objective	A Kernelized multi-subspace representation-based BSS approach is presented that allows the mixing process not only nonlinearity but underdetermined mixture.
Conventional Approaches	1. Underdetermined BSS (UBSS) [52, 53]: The separation criterion is on the assumption, that there are some TF points, where only one channel is dominant. – One technique to separate more sources than sensors is based on sparseness. 2. Harmeling et al. [42, 43, 44]: a kernelized TDSEP (KTDSEP) method was proposed for nonlinear blind source separation that the kernel functions are used for mapping the observations into the kernel spaces. 3. N-FINDR [54, 55]: the approach extract some geometric vertices of convex hull that are considered to form a set of basis
Conventional Approaches Limitations	1. The cost of storing and evalusting the model is proportinal to the number of data points [46]. 2. The method may fail if some sources lack specific time structure [46]. 3. The approach can not be used for the underdetermined problem.
Proposed Approach	1. The approach is to find a set of orthogonal basis that allow us to parameterize the signals in the multiple feature spaces with the reduced storage requirement. 2. To derive the coefficient matrix by solving the loss function on the coding coefficient vector.
Improvements and Discussion	1. Kernelized multi-subspace representation tackles the scenario of the nonlinear and underdetermined mixture. 2. The mapping function do not have any optimizing property in terms of the contrast function that allows them to be ranked and evaluated [45].

improper model construction. In this chapter, we work on the convergence and asymptotic analysis of the proposed separation approach in Chapter 3, where the nonlinearity of the

Table 1.4: Summary of Chapter 6.

Objective	ϵ-polynomial networks-based (ϵ-VPNs) BSS approach is presented that allows the mixing process not only nonlinearity but underdetermined mixture.
Conventional Approaches	1. Vanishing component analysis [56, 57]: The approach is inspired by ideas from the concept of vanishing idea, that gives the theorem support for existing of a finite set of polynomial. 2. Polynomial networks [58] The approach constructs deep neural networks, in which the output of each node is a quadratic function of its inputs.
Conventional Approaches Limitations	1. The approach can not be used for the underdetermined problem. 2. There is no clear-cut guidance on how one should choose the architecture and size of the network, or the type of computations it performs. Even when these are chosen, training these networks involves nonconvex optimization problems, which are often quite difficult.
Proposed Approach	1. Similar to the deep architectures, a novel polynomial networks is used to extend the linear BSS method to the nonlinear case. 2. The layers of our network start with polynomials of degree 1. To create the higher level representations of the data to decrease the bias, we next make the network deeper and deeper. 3. Once the polynomial networks are built, the coefficient matrix can be estimated by solving an optimal problem on the coding coefficient vector.
Improvements and Discussion	1. In particular, our network can search the number of layers that makes deeper until the candidate dataset becomes empty. 2. Since the approach exhausts all possibility candidate combination, the complexity and storage requirements of the model are proportional to the number of layers.

mixture function is extracted by the flexible approximation and the nonlinear problem is solved linearly in the feature space. The analysis stems from the performance of a mismatched estimator that accesses the finite sample size. By providing a closed-form expression of the mean squared error (MSE), we can present a novel algebraic formalization as

well as derive an upper bound on the estimation error. The simulation results show that if the nonlinearity of mixing functions can be extracted by the flexible approximation, the consistency of numerical MSE and analytical MSE can be achieved as the sample size tends to be infinity. This implies that the algorithm is feasible to separate the distortion of the nonlinear mixture.

- Chapter 5 works on the scenario when the mixing process is complex, such as the case where sources are mixed with some direction angles, or where the number of sensors is less than that of sources. We propose a Kernelized multi-subspace representation based BSS approach that allows the mixing process not only nonlinearity but underdetermined mixture. The approach relies on a Kernelized multi-subspace structure and sparse representation in the time-frequency (TF) domain. By parameterizing such subspaces, we can map the observed signals in the feature space with the coefficient matrix from the parameter space. We then exploit the linear mixture in the feature space that corresponds to the nonlinear mixture in the input space. Once such subspaces are built, the coefficient matrix can be constructed by solving an optimization problem on the coding coefficient vector. Relying on the TF representation, the target matrix can be constructed in a sparse mixture of TF vectors with the fewer computational cost. The experiments are designed on the observations generated from an underdetermined mixture and collected with some direction angles in a virtual room environment.

- In Chapter 6, similar to the deep architecture, a novel ϵ-vanishing polynomial networks (ϵ-VPNs) is proposed to extend the linear BSS method to the nonlinear and underdetermined case. The approach attempts to construct the ϵ-VPNs using some vanishing polynomials, so as to extract the nonlinearity or distortion caused by nonlinear mixing. Relying on such approximated bases are generated for the values attained by a set of mapping functions, we construct the architecture with increasing expressiveness, where the layer of our network begins with the polynomial of degree 1, up to build an output layer that can represent data with a small bias by a good approximate basis. Relying on several transformations of the input data, with higher-level representation from lower-level ones, the networks are to fulfill a mapping implicitly to the high-dimensional space. Once the ϵ-VPNs are built, we can fulfill a simple linear separation algorithm on top of this output as back propagation.

- Finally, Chapter 7 contains the conclusion and perspectives for future works.

1.3.2 Scope of the Dissertation

This dissertation consists on six chapters. In Chapters 3, 5 and 6, the novel models are presented to address the different problems related to recover the original sources from observations or mixture. Chapter 4 gives the convergence and asymptotic analysis for the proposed separation model in Chapter 3. Each chapter contains the problem formulation, the relevant existing literature, the proposed methods and their evaluation. The outline of this dissertation is summarized in Fig. 1.6.

Among the various linear separation methods shown in Fig. 1.5, the thesis discusses and uses some of them marked in blue. If the number of sources M is assumed to equal the number of sensors N, the mixing is referred to as determined ($N = M$). Overdetermined ($N > M$), if

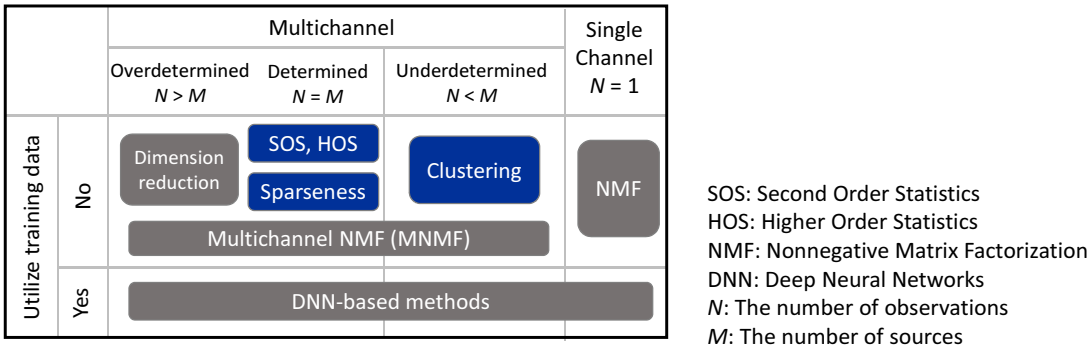


Figure 1.5: Various Separation Method for Linear BSS. Methods marked in blue are used for the further discussion in the later chapters.

the number of sources M is assumed to less than that of sensors N. The separation methods can be divided into methods based on higher order statistics (HOS), methods based on second order statistics (SOS), and the methods based on the sparseness. Based on the SOS by requiring only

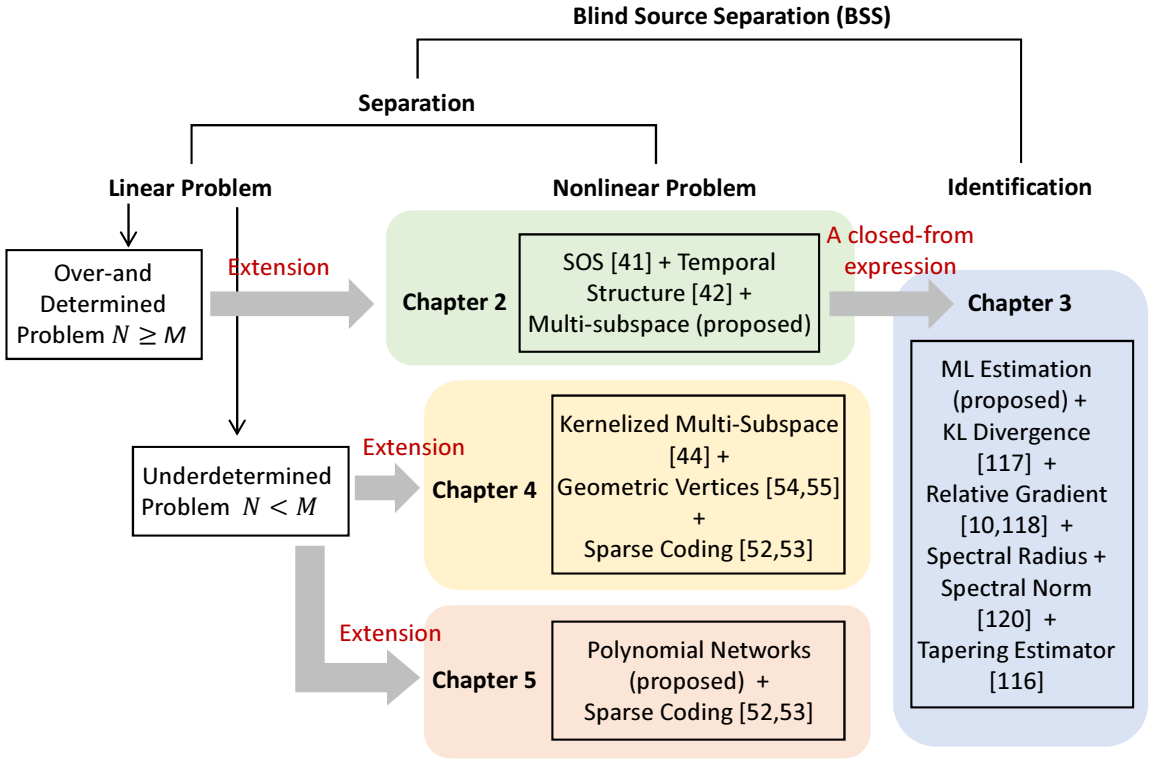


Figure 1.6: The Position in Existing Research of Each Chapter.

of the sources [59], or a minimum phase mixing system. The main advantage of SOS is that they are less sensitive to noise and outliers [60], and hence require less data for their estimation.

If the number of sources M is larger than that of sensors N, the mixing is referred to as underdetermined $(N < M)$. Clustering has also been introduced for underdetermined source separation [61, 62, 63, 64, 65]. If the sources are projected into a space where each source groups

together, the source separation problem can be solved with clustering algorithms. In [53, 66] the mask is underdetermined by clustering with respect to amplitude and delay differences.

A major advance in the field is to show how these frameworks can be extended to the nonlinear case. In this thesis, we mainly work on the relative extension of linear separation so that develop some practical model for nonlinear BSS, see Fig. 1.6. In Chapter 3, the proposed approach exploits on the extension of SOS and the temporal information of sources. To generates some feature spaces used to extract the nonlinearity of the mixing function, we provide a multi-subspace architecture. By mapping the data into the feature space, we can fulfill a simple SOS on top of these output to recover sources. In Chapter 4, a closed-form expression is derived to show the upper bound of the error of the proposed model in Chapter 3. Both Chapter 5 and Chapter 6 work on the underdetermined BSS problem from the extension of linear separation. Similar, by allowing multiple sources to be presented at any point in the TF domain, we can figure out the coefficient matrix in a sparse mixture TF vectors. The recovered sources thus can be derived by utilizing the coefficient matrix. The difference is that, in Chapter 5 the model relies on a Kernelized multi-subspace representation. The approach performs the nonlinear BSS by mapping data into the some kernel spaces. Key assumption is that approach generates some kernel feature spaces that are chosen enough to extract the nonlinearity of the mixing function with low computational cost. In contrast, in Chapter 6, the approach generates the approximation with increasing expressiveness, where the layer begins with polynomial of degree 1, up to build an output layer that can represent data with a small bias by a good approximate basis. Thus, the model in Chapter 6 exhibits a higher separation accuracy but high computational cost.

Chapter 2

State of the Art and Mathematical Preliminaries

This chapter introduces some mathematical preliminaries for blind source separation (BSS) of nonlinear mixing models. A fundamental difficulty in the nonlinear BSS problem is that it is highly non-unique without some extra constraints, which are often realized by using a suitable regularization. Contrary to the linear case, we consider it different from the respective nonlinear BSS problem. After considering these matters, some methods introduced for solving the nonlinear BSS problems are discussed in more detail. Special emphasis is to a multi-subspace mapping approach that applies ensemble learning to a flexible multilayer perceptron model for finding the sources and nonlinear mixing mapping that have most probably given rise to the observed mixed data. At the end of the chapter, other techniques introduced for solving the nonlinear mapping are reviewed.

2.1 Linear Instantaneous Mixtures

Blind source separation algorithms are based on different assumptions on the sources and the mixing system. In general, the sources are assumed to be independent or at least decorrelated. The separation criteria can be divided into methods based on higher order statistics (HOS), and methods based on second order statistics (SOS). Instead of spatial diversity a series of algorithms make strong assumptions on the statistics of the sources. For instance, they may require that sources do not overlap in the time-frequency domain, utilizing therefore a form of sparseness in the data. Similarly, some algorithms for acoustic mixtures exploit regularity in the sources such as

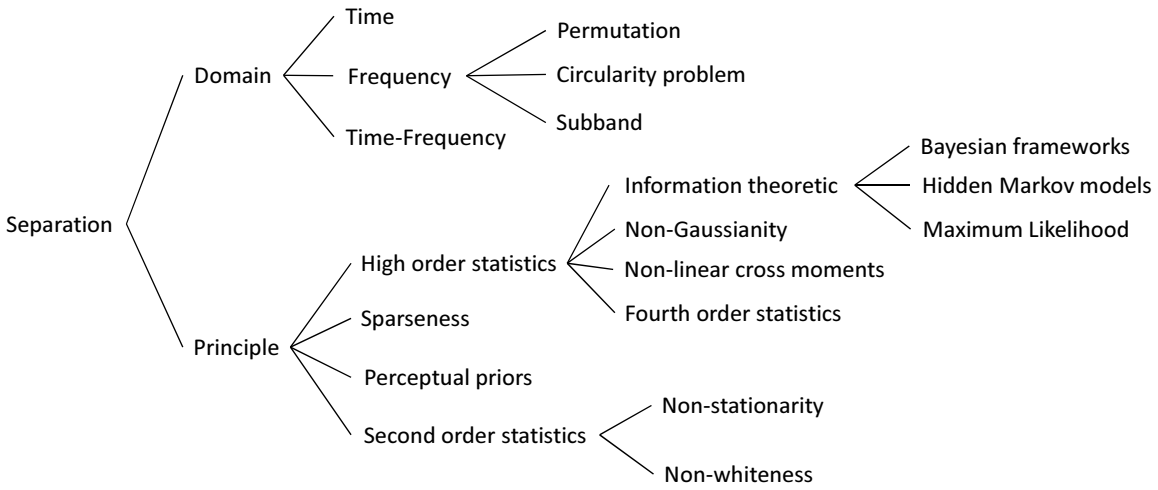


Figure 2.1: A Brief Overview of Important Areas within Blind Separation

The blind source separation can be divided into 3 classes, which are time domain, frequency domain and time-frequency domain. The second classification approach is based on the principle the author used for blind source separation. For this classification approach, the blind source separation can be divided into 4 classes, which are higher order statistics, sparseness, perceptual priors and auditory scene analysis, and second order statistics. Source separation based on higher order statistics is based on the assumption that the sources are statistically independent. Separation based on second order statistics requires only non-correlated sources rather than the stronger condition

of independence. The technique to separate more sources than sensors is based on sparseness. Some methods rely on insights gained from studies of the auditory system is based on the priors from auditory scene analysis and psycho-acoustics.

2.1.1 Domain

In Nishikawa et al. [67] the advantages and disadvantages of the time and frequency domain approaches have been compared. This is summarized in Table2.1.

Table 2.1: Advantages and Disadvantages for Separation in the Time or Frequency Domain.

Time Domain	
Advantages	Disadvantages
• The independence assumption holds better for full-band signals.	• Degradation of convergence in strong reverberant environment.
• Possible high convergence near the optimal point.	• Many parameters need to be adjusted for each iteration step.
Frequency Domain	
Advantages	Disadvantages
• The convolutive mixture can be transformed into instantaneous mixture problems for each frequency bin.	• For each frequency band, there is a permutation and a scaling ambiguity which needs to be solved.
• Due to the FFT, computations are saved compared to an implementation in the time domain.	• Problem with too few samples in each frequency band may cause the independence assumption to fail.
• Convergence is faster.	• Circular convolution deteriorates the separation performance.
	• Inversion of W is not guaranteed.

Algorithms that define a separation criteria in the time domain do typically not exhibit frequency permutation problems, even when computations are executed in the frequency domain [68] and [69]. A number of authors have therefore used time-domain criteria combined with frequency domain implementations that speed up computations. However, note that second-order criteria may be susceptible to the permutation problem even if they are formulated in the time domain [70].

2.1.2 Principle

Blind source separation algorithms are based on different assumptions on the sources and the mixing system. In general, the sources are assumed to be independent or at least decorrelated. The high order statistics (HOS) requiting the strong condition of independent. Instead of assumptions

on higher order statistics some methods make alternate assumptions of non-correlated, such as the non-stationarity of the sources [71], or a minimum phase mixing system [19].

Statistical independence between the source signals can also be expressed in terms of the probability density functions (PDF). If the model sources $\mathbf{x}$ are independent, the joint probability density function can be written as

$$p(\mathbf{x}) = \prod_n p(x_n). \tag{2.1}$$

Information theoretic methods for source separation are based on maximizing the entropy in each variable. The PDF is either assumed to have a specific form or it is estimated directly from the recorded data, leading to parametric and non-parametric methods respectively.

A kind of expression is a Bayesian formulation [72]. The advantage of a Bayesian formulation is that one can derive an optimal, possibly non-linear estimator of the sources enabling the estimation of more sources than the number of available sensors.

Traditional ways to solve the blind source separation problem only considers the non-Gaussian signal [73], without taking into account the time structure of the signal information. Generalized self-related and non-Gaussian source separation method are used to deal with the full account of the non-Gaussian signal and time structure information, to solve the blind source separation problem in the time structure.

Some algorithms [74] apply higher order statistics for separation of convolutive sources using nonlinear function by requiring the cross-moments between the two odd non-linear functions zero. The Taylor expansion of these functions captures higher order moments and this is found sufficient for separation of convolutive mixtures.

Non-stationary signals: Have also been developed algorithms for separating non-stationary signals. Matsuoka [75], for example, was the first to develop a method assuming that the energy ratio of two signals is a non-constant function of time and using the covariances matrix and a stochastic gradient method.

Numerous source separation applications are limited by the number of available microphones. It is in not always guaranteed that the number of sources is less than or equal to the number of sensors. With linear filters it is in general not possible to remove more than $M-1$ sources from the signal. By using nonlinear techniques, in contrast, it may be possible to extract a larger number of source signals. One technique to separate more sources than sensors is based on sparseness [76], [77] and [78]. If the source signals do not overlap in the time-frequency (T-F) domain it is possible to separate them.

2.2 Nonlinear BSS

While sources can be separated rather easily from a linear mixture (1.1), the corresponding problem with a nonlinear mixture

$$\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t)), \quad t = 1, 2, \cdots, T, \tag{2.2}$$

where $\mathcal{F} : \mathbb{R}^M \rightarrow \mathbb{R}^N$ is a nonlinear function, is significantly more difficult. Any potential solution is clearly non-unique due to possible undetermined scalar nonlinearies in the sources.

This follows from the fact that if random variables s_i and s_j are independent, so are $g_i(s_i)$ and $g_j(s_j)$ for any invertible $g_i, g_j : \mathbb{R} \to \mathbb{R}$ [79]. Unfortunately this is only the first of a list of indeterminacies.

2.2.1 Separability

In a sense, separating independent components with a nonlinear mapping is very simple, even too simple. In fact, any N-dimensional random vector $\mathbf{x}$ can be quite easily transformed nonlinearly to another N-dimensional random vector $\mathbf{y} = g(\mathbf{x})$ whose components are independent [29, 79]. This can be accomplished by a simple construction similar to the GramSchmidt orthogonalisation procedure.

In the construction, $\mathbf{y}$ can be assumed to have uniform density in the unit hypercube $[0, 1]^N$. This yields the condition

$$p(\mathbf{x}) = p_{\mathbf{y}}(\boldsymbol{g}(\mathbf{x}))|\det(D_{\boldsymbol{g}}(\mathbf{x}))|, \tag{2.3}$$

where $D_{\boldsymbol{g}}$ is the Jacobian matrix of the function $\boldsymbol{g}$. Looking for a solution of the form

$$g_i(\mathbf{x}) = g_i(x_1, x_2, \cdots, x_i), \quad i = 1, 2, \cdots, T, \tag{2.4}$$

the determinant of the Jacobian reduces to a product of terms $\partial g_i(\mathbf{x})/\partial x_i$. On the other hand, $p(\mathbf{x})$ can be decomposed as

$$\begin{aligned} p(\mathbf{x}) &= p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)\cdots, p(x_N|x_1, x_2, \cdots, x_{N-1}) \\ &= |\det D_{\boldsymbol{g}}(\mathbf{x})| = \prod_{i=1}^{N} \frac{\partial g_i(x_1, x_2, \cdots, x_i)}{\partial x_i} \end{aligned} \tag{2.5}$$

This is clearly satisfied if

$$\frac{\partial g_i(x_1, x_2, \cdots, x_i)}{\partial x_i} = p(x_i|x_1, x_2, \cdots, x_{i-1}), i = 1, 2, \cdots, T, \tag{2.6}$$

Integrating this yields a solution for g_i as the conditional cumulative density function of x_i given $x_1, x_2, \cdots, x_{i1}$, for all $i = 1, 2, \cdots, T$.

As can be seen, the above construction contains many arbitrary choices, such as the use of uniform density and the assumed form of $\boldsymbol{g}$. It is therefore not very surprising that the separation result is not at all unique, as shall be shown next.

2.2.2 Uniqueness

Recalling the definition of independence of the components of random vector $\mathbf{x}$, it is clearly preserved by mappings performing a permutation of the components and possibly some scalar transformations as in

$$\boldsymbol{g}(\mathbf{x}) = [g_1(x_{\sigma_{(1)}}), g_2(x_{\sigma_{(2)}}), \cdots, g_n(x_{\sigma_{(n)}})], \tag{2.7}$$

where $\sigma \in S_n$ is a permutation and $g_1, g_2, \cdots, g_n : \mathbb{R} \to \mathbb{R}$ are invertible scalar functions.

It can be shown [80] that mappings of the form (2.7) with invertible $g_1, \cdots, g_n$ are in fact the only invertible mappings that map all random vectors with independent components to random vectors with independent components. This does not mean that there would be no other such mappings for specific random vectors. This can be seen from the following construction for two uniformly distributed random variables [29].

Let x_1 and x_2 be independent random variables that are uniformly distributed on the interval $[0, 1]$, thus jointly uniformly distributed in the unit square $[0, 1] \times [0, 1]$. Any transformation g of the variables that preserves the volume does not alter the distribution of the variables and hence their independence. This happens if $|\det D_{\boldsymbol{g}}(\mathbf{x}) = 1|$ for all $\mathbf{x}$.

Volume preserving transformations of two variables are easy to represent by replacing the Cartesian coordinates x_1 and x_2 with polar coordinates r and θ specified by

$$x_1 = r\cos\theta, \quad x_2 = r\sin\theta. \tag{2.8}$$

A set of volume preserving transformations can now be defined by

$$r' = r, \quad \theta' \doteq \theta + f(r) \cdot \theta_0 (mod\ 2\pi), \tag{2.9}$$

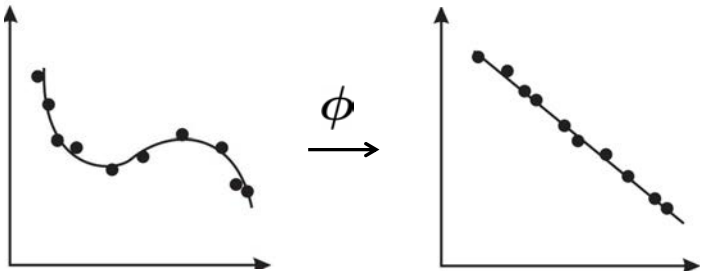
where $f(r)$ is a suitable scalar function and $\theta_0 \neq 0$ is a constant. Choosing, for instance, a smooth $f(r)$ with $f(r) = 0$ for $r > \frac{2}{3}$ and $f(r) = 1$ for $r < \frac{1}{3}$ provides a smooth transformation from x_1 and x_2 to another pair of independent random variables x_1' and x_2' that is not of the form (2.7). Condition $|\det D_{\boldsymbol{g}}(\mathbf{x}) = 1|$ can be easily verified to apply for this transformation.

This construction can be combined with the diagonalisation procedure to generate a class of nontrivial nonlinear mappings that are unrelated to each other, and each map the given random vector to one with independent components. This shows the non-uniqueness of nonlinear BSS, any random vector can be nonlinearly decomposed into independent components in several nontrivially related ways. In order to achieve blind nonlinear separation of sources, additional constraints are thus needed. The above constructions show that even constraints such as smoothness of the mixing or demixing mapping or knowing the actual source distributions are not enough to guarantee separation. In different approaches to nonlinear BSS, the actual constraints are typically implicitly defined by the model and methodology used.

2.3 General Nonlinear Models and Algorithms

Despite the ill-posed nature of the nonlinear BSS problem, there are several nonlinear BSS methods. This section is not intended as a thorough review of these methods. Wall and Amemiya [81] present a more complete review on traditional parametric statistical nonlinear models while the neural and BSS models are reviewed in [79].

In general, many of the models proposed in literature are only suited for very low-dimensional data and the methods are demonstrated with mildly nonlinear two-dimensional mixtures. The two-dimensional case is significantly simpler than higher dimensional ones and such methods are mostly not considered here.



(a) Input space (b) Feature space

Figure 2.2: The Basic Idea of Nonlinear Mapping. The mapping ϕ transforms the input data points (black dots) into a high-dimensional feature space, where they can be described by a linear model (straight solid line). The linear model found in feature space corresponds to a nonlinear model in the input space (curved solid line).

2.3.1 Classical Algorithms

The first neural network model for nonlinear factor analysis (FA) was proposed by [82] at the same time as the first general statistical models appeared. His model included two MLP networks, one for mapping $\mathbf{s} \rightarrow \mathbf{x}$ and one for $\mathbf{x} \rightarrow \mathbf{s}$. An optional dynamic extension included another multilayer perceptron (MLP) networks [11].

Another classical neural model for such a purpose are auto-associative MLP networks, that are MLP networks trained with input-output pairs $(\mathbf{x}, \mathbf{x})$. The number of neurons in a hidden layer is restricted to be smaller than the number of inputs and outputs, thus creating a bottleneck. The extracted nonlinear features can be retrieved from the values of the hidden neurons. With standard back-propagation this approach is very prone to overfitting and local minima, but more advanced learning methods such as flat minimum search can provide a method for nonlinear BSS [83, 84] through sparseness of the extracted features.

MLP networks are also used as a basis of the variational Bayesian nonlinear BSS method presented in this thesis. In case of the variational Bayesian method, the MLP is used to model only the generative mapping $\mathcal{F}$ from $\mathbf{s}$ to $\mathbf{x}$.

The MISEP method by [85, 86] is a generalization of the infomax method of linear ICA for nonlinear mixtures using an MLP network to model the nonlinearity. The source separation is supposedly based on the smoothness constraint provided by the MLP. While even mathematical C^∞-smoothness of the mapping is not sufficient for ensuring nonlinear separation in theory, the method does provide good separation results in several artificial examples as well as in a real nonlinear image mixture problem. These results are probably due to the fact that even though an MLP network with enough hidden neurons is a universal approximator [87, 88], networks with a limited number of hidden neurons produce more restricted mappings. In fact, a network with invertible square weight matrices is a sufficiently specialised structure to allow limited theoretical analysis [89].

Kernel methods have recently become a popular method of producing nonlinear counterparts for many linear statistical methods [90]. They work for any method based on second-order statistics that can be evaluated through inner products of the observation vectors. The kernel methods are based on transforming the data nonlinearly with a mapping $\Theta : \mathbb{R}^N \rightarrow \mathcal{F}$ to a high dimensional or even infinite-dimensional feature space F and performing the linear algorithm on the

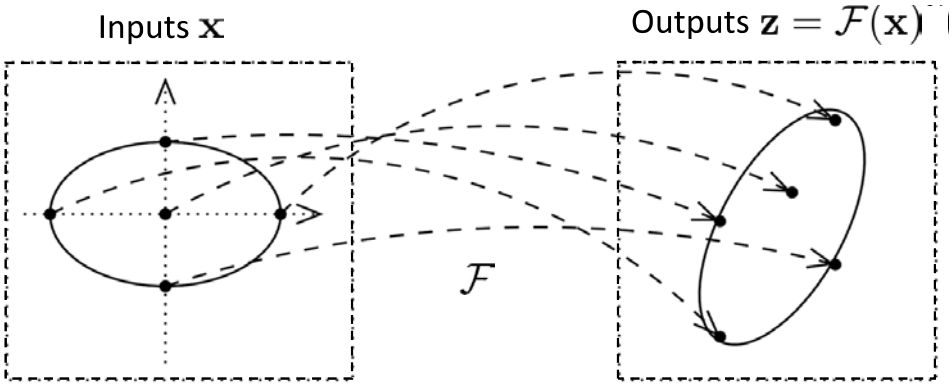


Figure 2.3: An Illustration of the Unscented Transform. The selected points $\mathbf{x}$ are mapped by $\mathcal{F}$ to $\mathbf{z}$ and the weighted mean and covariance of the points $\mathbf{z}$ are evaluated.

transformed data. This involves evaluating inner products of the transformed data vectors, but this can be done efficiently using the kernel trick of writing the inner product with the help of a kernel function k as

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = \Theta(\mathbf{x}) \cdot \Theta(\mathbf{y}). \tag{2.10}$$

This makes it easy to define, for instance, a kernelised version of the linear PCA algorithm [91]. The kernel PCA algorithm is used to aid the initialization of the variational Bayesian nonlinear BSS method.

ICA is inherently based on higher-order statistics and is therefore not directly kernelisable. Separation of temporally correlated sources is, however, possible using only second-order statistics. Harmeling et al. [44] propose a kernel method for nonlinear BSS of temporally correlated signals. The method is basically a kernelisation of the well-known TDSEP algorithm [41]. The problem with the method is the selection of the essential components from the multitude generated by the algorithm. The kernel based nonlinear BSS method should not be mixed with Kernel ICA, which is a method for separation of linear mixtures using contrast functions based on kernel methods [32].

The kernel BSS method is also closely related to the nonlinear version of independent slow feature analysis [45]. The nonlinear slow feature analysis method works by mapping the data nonlinearly to a high-dimensional feature space and looking for the slow components there. In its basic form, the method requires explicit expansion in the feature space and will thus probably not scale to large problems. Being mostly equivalent to the kernel TDSEP, it also suffers from the same problem of identifying the meaningful components.

2.4 Machine Learning Approaches

Both neural networks and kernel methods are universal function approximators (see for instance [92] and [93], respectively), which means that they can approximate a nonlinear mapping with any

given accuracy. However, neural networks usually require a high number of parameters, and their optimal configuration is found by performing an iterative nonlinear optimization process, often implemented through back-propagation. For a multitude of problems, this training procedure is slow, and it does not guarantee convergence to the optimal solution but rather encounters one of the multiple local minima [55].

Kernel methods, on the other hand, generally admit a more elegant solution which stems from the framework of RKHS and the convexity of the resulting optimization problem. Therefore, much kernel-based algorithms have a unique global solution that can be found by solving a convex optimization problem. As a result, although kernel methods are only a decade old, they now represent an established framework to solve machine learning problems and they are backed by an extensive list of experimental accomplishments. Some of the best known kernel methods are support vector machines (SVM) [94], kernel principal component analysis (kernel PCA) [95], kernel-based regression techniques [90], kernel canonical correlation analysis (kernel CCA) [32], kernel Fisher discriminant analysis (KFD) [96] and spectral clustering [97]. Successful applications of these algorithms have been reported in many fields, such as image processing, computational biology, bioinformatics, communications and medicine.

2.4.1 Other Existing Approximations

Traditional algorithms such as extended Kalman filtering are mostly based on the Taylor approximation [98]. The unscented transform and corresponding unscented Kalman filter were proposed by Julier and Uhlmann [99] to help avoid some of the problems of the Taylor approximation. The filter has since been further refined for instance by Wan et al. [100].

In a d-dimensional case, the unscented transform is based on selecting a set $\mathbf{x}$ of $2d$ weighted points together with the mean point that describe well the input distribution. In case of diagonal input covariance, the points will reside on the coordinate axes at a distance governed by corresponding standard deviation. These points are then transformed individually to get a new set of points $\mathbf{z}_i = \mathcal{F}(\mathbf{x}_i)$. The output mean and covariance are then computed as weighted mean and covariance of the transformed points $\mathbf{z}$. The procedure is illustrated in Fig. 2.3.

The unscented transform is intuitively appealing, but unfortunately it does not scale to high-dimensional problems and can even produce worse results than the Taylor approximation. The computational cost for the MLP case, which is linear in the total number of inputs and weights, can also get quite high when there are many sources.

Chapter 3

Nonlinear BSS Approach with Multi-Subspace Representation

This chapter describes the proposed method to tackle the ill-posed of the nonlinear blind source separation (BSS). An overview of BSS method and challenge in nonlinear BSS is introduced in the beginning of Chapter 3.1. Then, we summarize the the contribution of the proposed approach and some related works. The preliminary and problem formulation are given in Chapter 3.2. In Chapter 3.3, we present a novel approach used for nonlinear BSS algorithm and its analysis of properties. In Chapter 3.4, we discuss the computational cost of the proposed algorithm. Chapter 3.5 provides experimental results to illustrate the effectiveness of the proposed algorithm. We conclude the chapter briefly in Chapter 3.6.

3.1 Introduction

The problems of independent component analysis (ICA), blind separation of source signals have received wide attention in various fields such as speech enhancement [101], image recognition [102], wireless communication [103], and thus have been thoroughly studied in the signal processing community. Usually, the original sources are linearly or nonlinearly mixed in some ways to produce a number of observations. BSS aims at recovering independent sources from their mixtures having access only to the observations without any prior knowledge, i.e., neither the sources nor the mixing matrix is known. The foundation assumption for linear blind source separation is that the statistical independence of the sources is usually sufficient to constrain the demixing functions up to the trivial transformations such as permutation and scaling.

An obvious extension for the task of BSS is that the observations are assumed to be generated from a set of sources by a nonlinear, instantaneous and invertible function. Roughly, the blind source separation seeks to find the mixing function or its inverse, solely based on the assumption that the sources are statistically independent. However, the indeterminacies imposed by the nonlinear model are difficult to handle [23]. Without extra constraints, the solutions are non-unique and then it suffers from the inability to recover the sources such as scaling and permutation [29]. In fact, there is an infinite number of possible nonlinear decompositions of a random vector into independent components, and those decompositions are not similar to each other in any trivial way [23]. The recovery inconsistency has been tackled by adding further prior information directly in the model or as a regularization term in the optimization processing procedure.

Various attempts [104, 105, 24] have been proposed to provide a theoretical understanding for solving the nonlinear mixing. Despite such progress, there are still many important open problems and unexplored areas, particularly in the nonlinear spaces and systems. The captured nonlinear features are in fact growing at an enormous rate. That necessitates higher advancement of algorithms and methods to extract models, patterns, and knowledge from nonlinear mixing. For instance, the approach that captures the topology of the space from data points is represented in [106, 107]. Studying of various aspects of data geometry including manifold learning have been proposed in [108] .

One way relies on such a flexible approximation, including multi-layer perceptron (MLP) neural network [27, 30], which is employed for estimating the nonlinear separation transform function. By restricting the smoothness of the target transforming[1], MLP provides the regularized

[1]The function f is a smooth transformation if its derivatives of any order always exist and they are continuous.

solutions to ensure that nonlinear ICA leads to the sources separable. However, the example presented in [31] shows that the smoothness property is not a sufficient condition for this purpose. Hyvärinen and Pajunen [29] show conformal mapping[2] may helpful. Nonlinear ICA is able to estimate a separation mapping up to the rotation when the mapping functions are restricted to the set of conformal mapping. Unfortunately, the angle preservation conditions seem very restrictive [33]. In particular, it is not realistic in the framework of the nonlinear mappings associated with the nonlinear sensor array.

3.1.1 Our Contribution

We present a novel separation model that relies on the temporal structure and a novel mathematical construction with a multi-layer architecture. The approach pre-processes the data using a flexible approximation that projects the data into a high dimensional feature space. Then, by considering the temporal decorrelation as the separation criterion, we can break a nonlinear problem down into a version of the generalized joint diagonalization problem in the feature space.

The derivation of our algorithm is inspired by the idea of an efficient layer-by-layer representation to approximate such nonlinearity, which is referred to as Vanishing Ideal-based Non-Linear SEparation Model (ViNLisem). By using vanishing component analysis (VCA) in [56], a prominent work in machine learning, we generate a set of polynomial functions that transform a time-invariant nonlinear BSS to the local linear problem. Such transformed components are used to extract the nonlinear mixture as the flexible approximation. Similar to a well-known principle in modern deep learning [109, 110, 111], the layers of our architectures are built one-by-one, creating higher-and-higher level representations of the data. Once such a representation is built, a final output layer is constructed by solving a convex optimization problem [112]. Based on the multi-layered architecture, the nonlinearity of the mixing model is depicted by such polynomials. Importantly, the parameters and forms of polynomials depend solely on the input data, which guarantee the robustness of the structure. We thus address the general problem without being restricted to any specific mixture or parametric model.

In particular, the layer-by-layer representation is adaptively generated solely on the observations. As the number of spanned spaces goes up, the computational complexity grows exponentially. To overcome this obstacle, relying on the properties of vanishing components, we provide a feasible way to narrow the size of the candidate polynomial set. We thus generate the polynomial in the current layer only from the spanned space of the last layer and that of the first layer, such as $\boldsymbol{g}^{(t)}(\mathcal{S})$ is generated from the span of $F_{t-1} \times F_1$ rather than considering all the extended spaces, i.e., $F_1, F_2, \cdots, F_{t-1}$. The details are shown in Theorem 2 and Theorem 3.

In addition, using the frameworks in [41], the local temporal structure of the transformations is taken into account. The contrast function is discriminative to be designed by emphasizing the difference from the temporally i.i.d. data. On the other hand, the criterion is formulated by minimizing the second-order statistics in which the transformed components and their time lags are statistically as independent as possible. Therefore, we can break a nonlinear problem down into the version of generalized joint diagonalization problem in the feature space.

[2]The conformal mapping is defined as a mapping which preserves orientated angles. It is often considered in the framework of functions of complex-valued variables that are restricted to plane mapping. e.g., Joukovski mapping.

3.1.2 The Relative Work

One of the earliest frameworks based on temporal structures is Temporal Decorrelation source SEParation method, which is abbreviated as TDSEP in [41]. It works on the temporal structure that the separated signal and its time lags are jointly taken into account for the independence of the sources. However, for most temporal blind source separation (TBSS) methods, how to select the optimal time lags is an important problem. In this chapter, we are going to show how this framework can be extended to the nonlinear case rather than solving the problem of searching the optimal time lags.

A related but different idea is exploited in approximation using multi-kernel space. Harmeling et al. [42, 43, 44], a kernelized TDSEP (KTDSEP) method was proposed for nonlinear blind source separation that the kernel functions are used for mapping the observations into the kernel spaces. They show how kernel functions are employed to linear BSS methods to solve nonlinear source separation problems. These functions, however, do not have any optimizing property in terms of the contrast function that allows them to be ranked and evaluated. In addition, the method assumes the number of kernel spaces is chosen enough to approximate the nonlinearity without technical reasons. In [45], the authors claim that temporal slowness complements statistical independence well, and a combination of these principles leads to unique solutions of the nonlinear BSS problem.

Our construction and algorithm rely on the representation learning [113]. Heldt et al. [114] introduced a numerically stable approximate vanishing ideal algorithm. Livni et al. [58] defined a family of neural networks with polynomial activation functions that the polynomial features are learned as nonlinear combinations of the original signals. Donini and Aiolli [115], used a hierarchy of base kernel in the space of polynomial. These approaches consist of using an implicit map of the data, such as the Nyström method [116], random features [117] and sketching [118, 119]. That is features interactions in possibly high-dimensional data [120]. All of these approaches have in common with the flexible approximation, which emphasizes the representation learning as the key to the challenging nonlinear problems.

3.2 Preliminary and Problem Formulation

The nonlinear BSS problem is formally described as follows. The observed signals $\mathbf{x}(t) = \{x_1(t), x_2(t), \cdots, x_n(t)\}^\top$ are assumed to be generated from a set of statistically independent sources $\mathbf{s}(t) = \{s_1(t), s_2(t), \cdots, s_m(t)\}^\top$ by a nonlinear, instantaneous and invertible function

$$\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t)), \quad t = 1, 2, \cdots, T, \tag{3.1}$$

where $\{\cdot\}^\top$ denotes the transpose, and t is the sample (time) index. Here, T is the total number of time points. n and m refer to the number of observed signals and sources, respectively. In this chapter, we set $n = m$ in general. Since we are going to exploit only the statistical independence of the sources to be retrieved, a suitable approximation of the inverse nonlinear transformation could better reproduce the independence of the sources. Then some basic definitions are introduced for

problem setup. Let $\boldsymbol{f} \circ \boldsymbol{h}$ denotes the Hadamard product, such as $\boldsymbol{f} \circ \boldsymbol{h} = [f_1 h_1, f_2 h_2, \cdots, f_k h_k]^\top$, where $\boldsymbol{f} = \{f_1, f_2, \cdots, f_k\}$ and $\boldsymbol{h} = \{h_1 h_2, \cdots, h_k\}$ are two arbitrary vectors.

Definition 1 (Polynomial)**.** *A function g: $\mathbb{R}^n \to \mathbb{R}$ is called as polynomial if the linear combination is $g(\mathbf{x}) = \sum_j \beta_j \mathbf{x}^{\boldsymbol{\alpha}(j)}$, where the coefficient $\beta_j \in \mathbb{R}$, $\mathbf{x} = [x_1, x_2, \cdots, x_n]^\top$, $\mathbf{x}^{\boldsymbol{\alpha}(j)} \triangleq \prod_{i=1}^n x_i^{\alpha_i(j)}$ and $\boldsymbol{\alpha}(j) = [\alpha_1(j), \alpha_2(j), \cdots, \alpha_n(j)]^\top$.* □

Definition 2 (Polynomial Ring)**.** *The polynomial ring with n variables over $\mathbb{R}$ is denoted as $\mathbb{R}[x_1, \cdots, x_n]$ that the addition and multiplication operators over the polynomial ring are equivalent to addition and multiplication of functions.* □

Definition 3 (Ideal)**.** *Let I be a set of polynomials in $\mathbb{R}[x_1, x_2, \cdots, x_n]$, where $\mathbb{R}[x_1, x_2, \cdots, x_n]$ is a polynomial ring with n variables. For $\forall f \in I$ and $g \in \mathbb{R}[x_1, x_2, \cdots, x_n]$. If $fg \in I$ holds, then I is defined as an ideal.* □

Definition 4 (Set of Generators)**.** *Let I be an ideal. If $\forall f \in I$ there exist $h_1, h_2, \cdots, h_k \in \mathbb{R}[x_1, x_2, \cdots, x_n]$ and a set of polynomials $\{g_1, g_2, \cdots, g_k\} \subseteq I$ such that $f = \sum_i g_i h_i$, then $\{g_1, g_2, \cdots, g_k\}$ is said to generate I.* □

Definition 5 (Vanishing Ideal)**.** *Given a dataset $\mathcal{S} \subset \mathbb{R}^n$, for all $\mathbf{x} \in \mathcal{S}$, the vanishing ideal of $\mathcal{S}$ is the set of polynomials that vanish on $\mathcal{S}$. i.e. $g \in I(\mathcal{S})$ iff $g(\mathbf{x}) = 0$ for $\forall \mathbf{x} \in \mathcal{S}$.* □

The problem can be set up as follows. We have a set of observed signals $\mathcal{S} = \{\mathbf{x}(t)\}_{t=1}^T$ that are generated from (4.2). The objective is to estimate the original sources $\mathbf{s}(t)$ and the mixing functions $\mathcal{F}$ (or it's inverse function $\mathcal{G} = \mathcal{F}^{-1}$) by using the observed signals $\mathbf{x}(t)$ only.

However, without any extra constraints, the solutions of blind source separation are non-unique [29]. In this chapter, a novel approach is proposed by utilizing a flexible approximation to estimate the nonlinearity of the mixing function. First, let us focus on the representation learning [113]: how can we construct a structure that provides a good approximation basis for the values attained by vanishing polynomials.

Problem 1. *Given an input dataset $\mathcal{S} = \{\mathbf{x}(t)\}_{t=1}^T$, where $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^\top$ and $\mathcal{S} \subset \mathbb{R}^n$. The problem is to learn a set of vanishing polynomials, which is formulated as the following optimization problem*

$$\begin{aligned} &\min_{V} \quad \dim(V) \\ &\textit{subject to} \quad V = \{g_i(\mathbf{x}) = 0 \,|\, \mathbf{x} \in \mathcal{S}\}, \end{aligned} \tag{3.2}$$

for $i = 1, 2 \cdots, k$, where $\dim(\cdot)$ represents the dimension of variables and V denotes the set of vanishing polynomial. Since the real data are noisy that allow us to consider a tolerate value ϵ, such that the polynomials almost vanish on $\mathcal{S}$, i.e., $\|g_i(\mathbf{x})\| \leq \epsilon$ for $\forall \mathbf{x} \in \mathcal{S}$ is satisfied, where $\|\cdot\|$ denotes the Euclidean norm. □

In Problem 1, we prefer to seek a set of polynomials such that $g_i(\mathbf{x}) \approx 0$ for all i and $\mathbf{x} \in \mathcal{S}$. These polynomials may provide a sufficient characterization of elements in $\mathcal{S}$. By utilizing the generators of vanishing polynomials, any nonlinear mixture can be approximated with the combination of coefficients and the monomials. However, such polynomials did not achieve the inversion of the $\mathcal{F}$ function directly. They provide more features with a different selection of vanishing

polynomials. Finally, the sources are recovered by solving a joint diagonalization problem in the feature space.

The procedure is implemented by finding a set of polynomials $g_1(\mathbf{x})$, $g_2(\mathbf{x})$, $\cdots$, $g_k(\mathbf{x})$ that satisfy $\|g_i(\mathbf{x})\| \leq \epsilon$ for all $i = 1, \cdots, k$ and $\mathbf{x} \in \mathcal{S}$. Given a dataset $\mathcal{S}$, the vanishing ideal is denoted as $I(\mathcal{S})$, which is a set of polynomials vanished on $\mathcal{S}$, i.e., $g \in I(\mathcal{S})$ iff $\|g(\mathbf{x})\| \leq \epsilon$ for $\forall \mathbf{x} \in \mathcal{S}$. If a set of polynomials can generate $I(\mathcal{S})$, then this set of polynomials is referred to as a set of generators for $I(\mathcal{S})$. Hilbert basis theorem in [57] told us that a finite set of generators exists for any ideal. A finite set of generators of the ideal is an attractive mechanism for describing $I(\mathcal{S})$, since all the elements in $I(\mathcal{S})$ can be derived from this set of generators. Thus, the mixing function $\mathcal{F}$ can be approximated by finding such a finite set of generators, whose elements are named as vanishing polynomials.

Using the vanishing polynomials, the projected signals take the form of $\boldsymbol{\phi}(\mathbf{x}(t))$ that is the projection of $\mathbf{x}(t)$ in the high-dimensional feature space. The demixing process can be expressed by a linear combination of these projected signals in the following formulation.

Problem 2. *Let $\{\mathbf{x}(t)\}_{t=1}^{T}$ be a set of observed signals. There is a set of polynomials g_i such that $\{g_i(\mathbf{x}(t))\}_{i=1}^{k}$ form a basis of $\mathbb{R}^n$. By using such polynomials g_i, the projected data of $\mathbf{x}(t)$ in feature space denoted as $\boldsymbol{\phi}(\mathbf{x}(t)) = \{\phi_1(\mathbf{x}(t)), \phi_2(\mathbf{x}(t)), \cdots, \phi_k(\mathbf{x}(t))\}$. Since the original sources $\mathbf{s}(t)$ are mutually independent, there exist a coefficient matrix $\mathbf{W}$ so as to*

$$\arg\min_{\mathbf{W}} \sum_{i \neq j} \mathbf{W}_{i,:} \boldsymbol{\Sigma}_{\boldsymbol{\phi}} \mathbf{W}_{j,:}^{\top} + \sum_{i \neq j} \sum_{l=1}^{N} \mathbf{W}_{i,:} \boldsymbol{\Sigma}_{\boldsymbol{\tau}_l} \mathbf{W}_{j,:}^{\top}, \tag{3.3}$$

where $\mathbf{W}_{i,:}$ and $\mathbf{W}_{j,:}$ are the i-th and j-th row of matrix $\mathbf{W}$, respectively. The matrices $\boldsymbol{\Sigma}_{\boldsymbol{\phi}} = \mathbb{E}[\boldsymbol{\phi}(\mathbf{x}(t))\boldsymbol{\phi}(\mathbf{x}(t))^{\top}]$ and $\boldsymbol{\Sigma}_{\boldsymbol{\tau}_i} = \mathbb{E}[\boldsymbol{\phi}(\mathbf{x}(t))\boldsymbol{\phi}(\mathbf{x}(t+\tau_i))^{\top}]$ are defined as the covariance matrix of $\boldsymbol{\phi}(\mathbf{x}(t))$ and the covariance matrix with time lags τ_i, respectively. Thus, the signal is defined by

$$\tilde{s}_j(t) = \sum_{i=1}^{k} W_{ji} \phi_i(\mathbf{x}(t)), \tag{3.4}$$

for $j = 1, 2, \cdots, k$, where W_{ji} denotes the (j, i)-th element of the coefficient matrix $\mathbf{W}$. □

Problem 2 implies that if we build a set of vanishing components, which computes such k polynomials $g_1, g_2, \cdots, g_k$, then we can recover the signals $\tilde{\mathbf{s}}(t)$ with k dimensions. Due to $k > n$, we need to select n sources from $\tilde{\mathbf{s}}(t)$, which construct the estimation of the original sources $\mathbf{s}(t)$.

Problem 3. *Let $\tilde{\mathbf{s}}(t) = [\tilde{s}_1(t), \tilde{s}_2(t), \cdots, \tilde{s}_k(t)]^{\top}$ be a set of recovered signals. Since the original sources $\mathbf{s}(t)$ are mutually independent, it is also independent if the separation process in (3.3) is applied again to the signal $\tilde{\mathbf{s}}(t)$. i.e., $\tilde{\mathbf{s}}'(t) = \mathbf{W}'\tilde{\mathbf{s}}(t)$, where $\mathbf{W}'$ is another coefficient matrix if joint diagonalization approach is applied to the signal $\tilde{s}(t)$ again. Then, the recovered sources $\hat{\mathbf{s}}(t)$ corresponds to the first n maximum correlations (*corr*) in $\tilde{\mathbf{s}}(t)$*

$$\begin{aligned} & \hat{\mathbf{s}}(t) = \tilde{\mathbf{s}}_{\boldsymbol{\pi},:}(t), \;\; t = 1, 2, \cdots, T, \\ \textit{subject to} \quad & \boldsymbol{\pi} = \Upsilon(\boldsymbol{\theta}; n), \\ & \boldsymbol{\theta} = \Xi_{\max} \left\{ \operatorname{corr} \left(\tilde{\mathbf{s}}(t), \tilde{\mathbf{s}}'(t) \right) \right\}, \end{aligned} \tag{3.5}$$

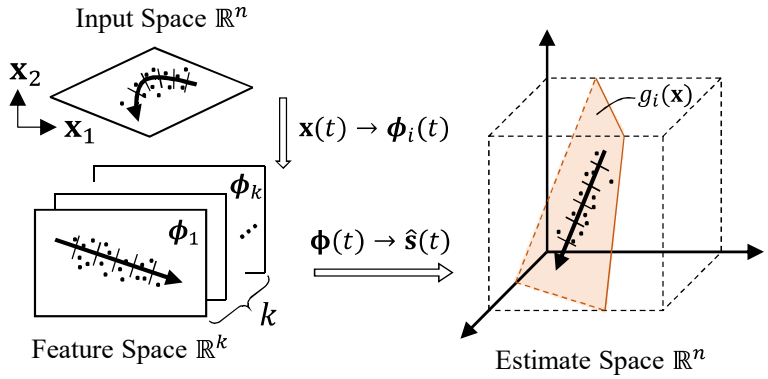


Figure 3.1: The Configuration of the Proposed Algorithm. Input data $\mathbf{x}(t)$ are mapped to the manifold of $\mathcal{G} \in \mathbb{R}^k$, which is a feature space constructed by some polynomials $\{g_1, g_2, \cdots, g_k\} \subset \mathcal{G}$. Therefore, the projected points $\boldsymbol{\phi}(\mathbf{x}(t))$ in feature space can make the problem linearly separable. The linear coefficient matrices in the feature space correspond to nonlinear coefficient matrices in the input space.

where $\tilde{\mathbf{s}}_{\boldsymbol{\pi},:}(t)$ *is the vector composed of elements from* $\tilde{\mathbf{s}}(t)$ *indicated with the index* $\boldsymbol{\pi}$. $\boldsymbol{\pi}$ *is the index number of output of* $\Upsilon(\boldsymbol{\theta}; n)$ *that is the function to choose the maximum* n *values of vector* $\boldsymbol{\theta}$. $\Xi_{\max}\{\mathrm{corr}\,(\tilde{\mathbf{s}}(t), \tilde{\mathbf{s}}'(t))\}$ *is a function to output a vector* $\boldsymbol{\theta}$ *with each element being as the maximum value of each row of the matrix* $\mathrm{corr}\,(\tilde{\mathbf{s}}(t), \tilde{\mathbf{s}}'(t))$.

Fig. 3.1 shows an intuitive example for nonlinear separation using the mapping functions. Since the observations are nonlinearly mixed in the input space, we need to resort to a flexible approximation that can extract the nonlinear characteristics in the manifold $\mathcal{G}$. Here, vanishing components allow us to construct the nonlinear variants by some polynomials, such as $g_1(\mathbf{x}(t)), g_2(\mathbf{x}(t)), \cdots, g_k(\mathbf{x}(t)) \in \mathcal{G}$. *i.e.*, the data $\mathbf{x}(t)$ are mapped implicitly into the feature space that denoted as $\boldsymbol{\phi}(\mathbf{x}(t)) = [\phi_1(\mathbf{x}(t)), \phi_2(\mathbf{x}(t)), \cdots, \phi_k(\mathbf{x}(t))]^\top = [g_1(\mathbf{x}(t)), g_2(\mathbf{x}(t)), \cdots, g_k(\mathbf{x}(t))]^\top$. The feature space is spanned from such polynomials that enable us to work on $\mathcal{G}$. Then BSS approaches can be applied to the projected data in the feature space, which corresponds to the nonlinear BSS approaches in the input space. Finally, due to $k > n$, we need to select n sources, which construct the estimation of the original sources $\mathbf{s}(t)$ in the estimated space. Since the parameters of the polynomials depend solely on the input data, it guarantees the robustness of the structure.

3.3 Nonlinear Separation Model

We now turn to develop our nonlinear separation model as well as the accompanying analysis. We do the algorithm in the following stages. First, we derive a flexible approximation with multi-layer architecture, which runs in a set of polynomials that approximately equal to the value of zero. Thus the projected data in the feature space can make the problem linearly separable. Then, by taking into account the temporal structure served as a separation criterion, we can break the nonlinear problem down into a joint diagonalization problem in the feature space.

3.3.1 Structure of Multi-Layer Architecture

In order to perform a simple linear separation problem in feature space that corresponds to the nonlinear problem in input space, we need to specify how to map inputs $\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T) \in \mathbb{R}^n$ into the feature space $\mathbb{R}^k$. A similar way is the kernel-based TDSEP presented by Harmeling et al. [44]. The difference is that our proposed method adapts to generate the polynomials, rather than assuming the number of approximate functions is chosen enough to represent the nonlinearity.

To ensure that a set of generators of $I(\mathcal{S})$ carry significant information about the input, we require the generators to be uncorrelated and the coefficients being in the null space of the matrix, which is composed of the monomials with different degree. Mathematically, this can be stated as follows.

Proposition 1. *Denote the set of monomials over n variables with total degree up to d by $\mathcal{T}_d^n$. Consider the set of monomials $\mathcal{T}_d^n$ and the matrix $\mathbf{A}$ of size $T \times |\mathcal{T}_d^n|$ as follows: $\mathbf{A}_{ij} = t_j(\mathbf{x}(i))$, where $t_j(\mathbf{x}(i))$ is the j^{th} monomials in $\mathcal{T}_d^n$, which is composed of elements from $\mathbf{x}(i)$. Let $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_k$ be a basis of the null space of matrix $\mathbf{A}$. Namely, for all $i = 1, 2, \cdots, k$, we have $\mathbf{A}\boldsymbol{\beta}_i = \mathbf{0}$ and any vector $\boldsymbol{\beta}$ that satisfies $\mathbf{A}\boldsymbol{\beta} = \mathbf{0}$ can be written as a linear combination of $\boldsymbol{\beta}_i$. Then the polynomials $f_i(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} \beta_{ij} t_j(\mathbf{x}), i = 1, 2, \cdots, k$ form a set of generators of $I(\mathcal{S})$, where β_{ij} is the coefficient for the i-th polynomial function and j-th monomial.* □

Proof. Since $\mathbf{A}\boldsymbol{\beta}_i = \mathbf{0}$ is satisfied for all $i = 1, 2, \cdots, k$, we have $f_i(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} \beta_{ij} t_j(\mathbf{x}) = 0$. Thus, $f_i(\mathbf{x}) \in I(\mathcal{S})$. Consider any polynomial $g(\mathbf{x})$ in the set of $I(\mathcal{S})$. Denote the coefficients for the polynomial $g(\mathbf{x})$ by $\mathbf{z} \in \mathbb{R}^{|\mathcal{T}_d^n|}$ such that the coefficients satisfy $\mathbf{A}\mathbf{z} = \mathbf{0}$. Then we have $g(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} z_j t_j(\mathbf{x}) = 0$. Since $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_k$ is a basis of the null space of matrix $\mathbf{A}$, the coefficient vector $\mathbf{z}$ can be written as a linear combination of $\boldsymbol{\beta}_i$ as $\mathbf{z} = \sum_{i=1}^{k} \alpha_i \boldsymbol{\beta}_i$, which we also have $z_j = \sum_{i=1}^{k} \alpha_i \boldsymbol{\beta}_{ij}$. Then the polynomial $g(\mathbf{x})$ can be written by $g(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} z_j t_j(\mathbf{x}) = \sum_{j=1}^{|\mathcal{T}_d^n|} \sum_{i=1}^{k} \alpha_i \boldsymbol{\beta}_{ij} t_j(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i f_i(\mathbf{x})$. Thus, the polynomials $f_i(\mathbf{x})$ form a set of generators of $I(\mathcal{S})$. □

The above procedure achieves the goal of finding a set of generators of $I(\mathcal{S})$. Since the real data are noisy that allow us to consider a tolerate value ϵ, such that the polynomials almost vanish on $\mathcal{S}$ if $\boldsymbol{g}(\mathbf{x}) \leq \epsilon$ is satisfied.

Polynomials of Degree 1

If the vanishing polynomial is applied to the whole data $\mathcal{S}$, we have

$$\boldsymbol{g}^{(1)}(\mathcal{S}) = \left[g^{(1)}(\mathbf{x}(1)), \mathbf{x}(2)), \cdots, g^{(1)}(\mathbf{x}(T))\right]^\top = \mathbf{0}_{T \times 1}, \tag{3.6}$$

where $\mathcal{S} = \{\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T)\}$. Firstly, the linear polynomial can be expressed as the combination of vector $\mathbf{x}(t)$ with the coefficient $\boldsymbol{\beta} \in \mathbb{R}^{n+1}$ such that

$$g^{(1)}(\mathbf{x}(t)) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i(t) = \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(t)), \tag{3.7}$$

Algorithm 1 Generate Polynomials of Degree 1 by Gram-Schmidt Procedure

Initialization:

$F_1 = \{\boldsymbol{\rho}_0(\mathcal{S})\}$, where $\boldsymbol{\rho}_0(\mathcal{S}) = [1/\sqrt{n}, 1/\sqrt{n}, \cdots, 1/\sqrt{n}]^\top$;
$V_1 = \emptyset$;
$C_1 = \{\boldsymbol{\rho}_1(\mathcal{S}), \boldsymbol{\rho}_2(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})\}$, where $\boldsymbol{\rho}_i(\mathcal{S}) = [\rho_i(\mathbf{x}(1)), \rho_i(\mathbf{x}(2)), \cdots, \rho_i(\mathbf{x}(T))]^\top$.

for $i = 1$ to n **do**
  $\boldsymbol{g}_i^{(1)}(\mathcal{S}) = \boldsymbol{\rho}_i(\mathcal{S}) - \sum_{\boldsymbol{\rho} \in F_1} \langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}(\mathcal{S}) \rangle \boldsymbol{\rho}(\mathcal{S})$
  if $\|\boldsymbol{g}_i^{(1)}(\mathcal{S})\| \leq \epsilon$ **then**
    $V_1 \leftarrow V_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{S})\}$
  else
    $F_1 \leftarrow F_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{S}) / \|\boldsymbol{g}_i^{(1)}(\mathcal{S})\|\}$
  end if
end for

Output:

Vanishing polynomial set V_1;
Non-vanishing polynomials set F_1.

where $x_i(t)$ is the i-th element for the observations $\mathbf{x}(t)$ and $\rho_i(\mathbf{x}(t)) = x_i(t)$ for convenience. Thus, $\rho_0(\mathbf{x}(t)) = 1$ for all $\mathbf{x}(t)$. It follows that for any such polynomial we have

$$\boldsymbol{g}^{(1)}(\mathcal{S}) = \begin{bmatrix} g^{(1)}(\mathbf{x}(1)) \\ g^{(1)}(\mathbf{x}(2)) \\ \vdots \\ g^{(1)}(\mathbf{x}(T)) \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n \beta_i \rho_i(\mathbf{x}(1)) \\ \sum_{i=0}^n \beta_i \rho_i(\mathbf{x}(2)) \\ \vdots \\ \sum_{i=0}^n \beta_i \rho_i(\mathbf{x}(T)) \end{bmatrix} = \sum_{i=0}^n \beta_i \boldsymbol{\rho}_i(\mathcal{S}), \tag{3.8}$$

where $\boldsymbol{\rho}_i(\mathcal{S}) = [\rho_i(\mathbf{x}(1)), \rho_i(\mathbf{x}(2)), \cdots, \rho_i(\mathbf{x}(T))]^\top$.

Theorem 1. *The polynomial $\boldsymbol{g}^{(1)}(\mathcal{S})$ vanishes on dataset $\mathcal{S}$ if and only if $\boldsymbol{g}^{(1)}(\mathcal{S}) = \mathbf{0}_{T\times 1}$, which requires the vector $\boldsymbol{\beta}$ would be in the null space of the $T\times(n+1)$ matrix $\mathbf{A}_1 = [\boldsymbol{\rho}_0(\mathcal{S}), \boldsymbol{\rho}_1(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})]$ as*

$$\mathbf{A}_1 \boldsymbol{\beta} = [\boldsymbol{\rho}_0(\mathcal{S}), \boldsymbol{\rho}_1(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})]\boldsymbol{\beta} = \mathbf{0}_{T\times 1}. \tag{3.9}$$

Then the vanishing polynomials can be obtained by searching the null space of $\mathbf{A}_1$. We maintain two sets for polynomials of degree 1: V_1 for the vanishing polynomials and F_1 for the non-vanishing polynomials. We use the notation $F_1 = \{\boldsymbol{\rho}(\mathcal{S}) : \boldsymbol{\rho} \in F_1\} \subset \mathbb{R}^T$ to denote the vectors in $\mathbb{R}^n$. We will construct F_1 such that F_1 is a set of orthogonal vectors in $\mathbb{R}^T$. Algorithm 1 describes the procedure to generate the vanishing and non-vanishing polynomials of degree 1 by the Gram-Schmidt procedure.

Considering a polynomial of degree 0, $\boldsymbol{\rho}_0(\mathcal{S}) = \mathbf{1}_{T\times 1}$ is clearly non-vanishing. We initialize $F_1 = \{\boldsymbol{\rho}_0(\mathcal{S})/\|\boldsymbol{\rho}_0(\mathcal{S})\|\}$, where $\|\cdot\|$ denotes the norm of the vector. And set $V_1 = \emptyset$ initially. Set C_1 to be a candidate set of polynomials, which is composed of polynomials of degree 1, such as $C_1 = \{\boldsymbol{\rho}_1(\mathcal{S}), \boldsymbol{\rho}_2(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})\}$. To obtain the non-vanishing polynomials orthogonal to each

other, it requires

$$\boldsymbol{g}_i^{(1)}(\mathcal{S}) = \boldsymbol{\rho}_i(\mathcal{S}) - \sum_{\boldsymbol{\rho}\in F_1} \langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}(\mathcal{S})\rangle \boldsymbol{\rho}(\mathcal{S}). \tag{3.10}$$

Since the non-vanishing polynomial set F_1 only contains one element $\boldsymbol{\rho}_0(\mathcal{S})$ initially, the above equation can be simply represented as

$$\boldsymbol{g}_i^{(1)}(\mathcal{S}) = \boldsymbol{\rho}_i(\mathcal{S}) - \langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}_0(\mathcal{S})\rangle \boldsymbol{\rho}_0(\mathcal{S}), \tag{3.11}$$

where $\langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}_0(\mathcal{S})\rangle$ is the coefficient for $\boldsymbol{\rho}_0(\mathcal{S})$. We can now reformulate (3.11) in terms of a dual representation as

$$\boldsymbol{g}_i^{(1)}(\mathcal{S}) = \Big[\boldsymbol{\rho}_0(\mathcal{S}), \boldsymbol{\rho}_1(\mathcal{S}), \cdots, \boldsymbol{\rho}_i(\mathcal{S}), \cdots, \boldsymbol{\rho}_n(\mathcal{S})\Big]^\top \begin{bmatrix} -\langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}_0(\mathcal{S})\rangle \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.12}$$

Compared with (6.10) in Theorem 1, the vector $\boldsymbol{\beta}$ is given in the form $\boldsymbol{\beta} = [-\langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{\rho}_0(\mathcal{S})\rangle, 0, \cdots, 1, \cdots, 0]^\top$. If a proper coefficient vector $\boldsymbol{\beta}$ can be searched so as to $\boldsymbol{g}_i(\mathcal{S})$ vanish on the data $\mathcal{S}$, we update $V_1 \leftarrow V_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{S})\}$. Otherwise, $F_1 \leftarrow F_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{S})/\|\boldsymbol{g}_i^{(1)}(\mathcal{S})\|\}$ is updated, where the normalization ensures that all the vectors in F_1 are orthonormalization as the normalized vectors. At the end of this process, F_1 contains a set of linear polynomials which are non-vanishing on $\mathcal{S}$ and V_1 contains a set of linear polynomials that vanish on $\mathcal{S}$.

Polynomials of Degree 2

To exploit the polynomials of degree 2, we need to construct a candidate set of polynomials $C_2 = \{\boldsymbol{\rho}_{i,j}(\mathcal{S})\}_{i,j=1}^n$, where $\boldsymbol{\rho}_{i,j}(\mathcal{S}) = [\rho_{i,j}(\mathbf{x}(1)), \rho_{i,j}(\mathbf{x}(2)), \cdots, \rho_{i,j}(\mathbf{x}(T))]^\top$ and $\rho_{i,j}(\mathbf{x}(t)) = x_i(t)x_j(t)$ for all i, j. Each polynomial of degree 2 takes the form

$$g^{(2)}(\mathbf{x}(t)) = \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(t)) + \sum_{i,j=1}^{n} \beta_{i,j}\rho_{i,j}(\mathbf{x}(t)). \tag{3.13}$$

By considering all the data points in $\mathcal{S}$, we have

$$\begin{aligned} \boldsymbol{g}^{(2)}(\mathcal{S}) &= \Big[g^{(2)}(\mathbf{x}(1)), \cdots, g^{(2)}(\mathbf{x}(T))\Big]^\top \\ &= \begin{bmatrix} \sum_{i=0}^n \beta_i\rho_i(\mathbf{x}(1)) + \sum_{i,j=1}^n \beta_{i,j}\rho_{i,j}(\mathbf{x}(1)) \\ \sum_{i=0}^n \beta_i\rho_i(\mathbf{x}(2)) + \sum_{i,j=1}^n \beta_{i,j}\rho_{i,j}(\mathbf{x}(2)) \\ \vdots \\ \sum_{i=0}^n \beta_i\rho_i(\mathbf{x}(T)) + \sum_{i,j=1}^n \beta_{i,j}\rho_{i,j}(\mathbf{x}(T)) \end{bmatrix} \end{aligned}$$

$$= \sum_{i=0}^{n} \beta_i \boldsymbol{\rho}_i(\mathcal{S}) + \sum_{i,j=1}^{n} \beta_{i,j} \boldsymbol{\rho}_{i,j}(\mathcal{S}). \tag{3.14}$$

As before, we can find vanishing 2^{nd} order polynomials via the null space of the matrix: $\mathbf{A}_2 = [\mathbf{A}_1, \boldsymbol{\rho}_{1,1}(\mathcal{S}), \boldsymbol{\rho}_{2,2}(\mathcal{S}), \cdots, \boldsymbol{\rho}_{n,n}(\mathcal{S})]$. To find the null space of the matrix $\mathbf{A}_2$, we could simply continue the Gram-Schmidt procedure that we have already performed for the columns of $\mathbf{A}_1$. However, we now need to consider $n^2 + n + 1$ columns. As the degree goes up, the number of columns increases exponentially. To overcome this obstacle, relying on the properties of vanishing components, we provide an effective iterative approach to narrow the size of the candidate polynomial set.

Theorem 2. *Let $\boldsymbol{g}^{(2)}(\mathcal{S})$ be a set of polynomials of degree 2. It can be constructed by two terms of degree 1 of the form $\boldsymbol{g}^{(2)}(\mathcal{S}) = \sum_{i_1,i_2} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)}$. Without loss of generality, assume that for $i_1, i_2 \leq l$, where l is index number of polynomial of degree 1. We have that both $\boldsymbol{f}_{i_1}^{(1)}$ and $\boldsymbol{f}_{i_2}^{(1)}$ are non-vanishing on $\mathcal{S}$. For $i_1, i_2 > l$, either $\boldsymbol{f}_{i_1}^{(1)}$ or $\boldsymbol{f}_{i_2}^{(1)}$ vanishes. It follows that for all $i_1, i_2 > l$ we have that the polynomial $\boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} = \mathbf{0}_{T\times 1}$. Thus, the polynomial $\hat{\boldsymbol{g}}^{(2)}(\mathcal{S}) = \sum_{i_1,i_2\leq l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)}$ satisfies $\hat{\boldsymbol{g}}^{(2)}(\mathcal{S}) = \boldsymbol{g}^{(2)}(\mathcal{S})$. $F_1 = \{\mathbf{p}_1^{(1)}, \mathbf{p}_1^{(2)}, \cdots, \mathbf{p}_{|F_1|}^{(1)}\}$ is denoted as a non-vanishing polynomial set of degree 1, where $|F_1|$ denotes the number of elements included in the set F_1. Any polynomial of degree 1 that generated from F_1 can be expressed as*

$$\boldsymbol{f}_{i_1}^{(1)} = \sum_{j_1} \alpha_{i_1,j_1}^{(1)} \mathbf{p}_{j_1}^{(1)}, \quad \boldsymbol{f}_{i_2}^{(1)} = \sum_{j_2} \alpha_{i_2,j_2}^{(1)} \mathbf{p}_{j_2}^{(1)}, \tag{3.15}$$

where $\alpha_{i_1,j_1}^{(1)}$ and $\alpha_{i_2,j_2}^{(1)}$ denote the coefficients that make $\boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} \neq \mathbf{0}_{T\times 1}$ for all $i_1, i_2 \leq l$. Then F_2 can be generated from the span of $\boldsymbol{f}_{i_1}^{(1)}$ and $\boldsymbol{f}_{i_2}^{(1)}$ for $i_1, i_2 \leq l$ as

$$\hat{\boldsymbol{g}}^{(2)}(\mathcal{S}) = \sum_{i_1,i_2\leq l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} = \sum_{j_1,j_2} \left[(\mathbf{p}_{j_1}^{(1)} \circ \mathbf{p}_{j_2}^{(1)}) \Big(\sum_{i_1,i_2\leq l} \alpha_{i_1,j_1}^{(1)} \alpha_{i_2,j_2}^{(1)} \Big) \right]. \tag{3.16}$$

The operator $\circ$ denotes the Hadamard product, namely the vector $\boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} = [f_{i_1,1}^{(1)} f_{i_2,1}^{(1)}, f_{i_1,2}^{(1)} f_{i_2,2}^{(1)}, \cdots, f_{i_1,T}^{(1)} f_{i_2,T}^{(1)}]^\top$, where the degree of $\boldsymbol{f}_{i_1}^{(1)} = [f_{i_1,1}^{(1)}, f_{i_1,2}^{(1)}, \cdots, f_{i_1,T}^{(1)}]^\top$ and $\boldsymbol{f}_{i_2}^{(1)} = [f_{i_2,1}^{(1)}, f_{i_2,2}^{(1)}, \cdots, f_{i_2,T}^{(1)}]^\top$ are at most 1. □

Theorem 2 is proved in the Appendix A. It follows that $\hat{\boldsymbol{g}}^{(2)}(\mathcal{S})$ can be constructed from the span of $F_1 \times F_1$ and thus to construct F_2 and V_2, which suffices to find the null space and range on the set of candidate polynomials from $F_1 \times F_1$. Formally, let us redefine C_2 to be the set

$$C_2 = \left\{ \boldsymbol{\rho}_{i_1,i_2\leq l}(\mathcal{S}) = \mathbf{p}_{j_1}^{(1)} \circ \mathbf{p}_{j_2}^{(1)} |\ \mathbf{p}_{j_1}^{(1)}, \mathbf{p}_{j_2}^{(1)} \in F_1 \right\}. \tag{3.17}$$

We will construct F_2 and V_2 by continuing a similar process with a polynomial of degree 1 on the candidate vectors of C_2. Note that, due to the particular structure of vanishing polynomials, as proposed in Theorem 2, $\boldsymbol{g}^{(2)}(\mathcal{S})$ can be generated from the span of $F_1 \times F_1$, i.e., from the polynomials with $i_1, i_2 \leq l$ rather than the whole candidate vectors. Therefore, the remainder of

$\boldsymbol{\rho}_{i_1,i_2\leq l} \in C_2$ after projecting it on the current set F_2 is the polynomial $\boldsymbol{g}^{(2)}(\mathcal{S})$ defined by

$$\boldsymbol{g}^{(2)}(\mathcal{S}) = \boldsymbol{\rho}_{i_1,i_2\leq l}(\mathcal{S}) - \sum_{\mathbf{p}^{(1)}\in F_2} \langle \boldsymbol{\rho}_{i_1,i_2\leq l}(\mathcal{S}), \mathbf{p}^{(1)}(\mathcal{S})\rangle \mathbf{p}^{(1)}(\mathcal{S}). \tag{3.18}$$

It requires $|F_1|\times|F_1|$ times to evaluate all the polynomials in the candidate polynomial set C_2. Before we evaluate the polynomials of degree 2, we initialize F_2 and V_2 as $F_2 = F_1$ and $V_2 = V_1$. Then if $|\boldsymbol{g}^{(2)}(\mathcal{S})| \leq \epsilon$, we have $\boldsymbol{g}^{(2)}(\mathcal{S})$ vanishes on $\mathcal{S}$. So we update $V_2 \leftarrow V_2 \bigcup \{\boldsymbol{g}^{(2)}(\mathcal{S})\}$. Otherwise, we update $F_2 \leftarrow F_2 \bigcup \{\boldsymbol{g}^{(2)}(\mathcal{S})/\|\boldsymbol{g}^{(2)}(\mathcal{S})\|\}$. At the end of this process, F_2 contains a set of polynomials of degree 1 and degree 2 that are non-vanishing on $\mathcal{S}$. In contrast, V_2 contains a set of polynomials of degree 1 and degree 2 that vanish on $\mathcal{S}$.

Polynomials with a Higher Degree

The above progress continues to a higher degree. For any polynomial of degree t, we prefer to construct the set of non-vanishing polynomials F_t only from the span of $F_{t-1}\times F_1$. At iteration t, the candidate polynomial set C_t is given in the form

$$C_t = \left\{ \boldsymbol{\rho}_{i_1,i_2,\cdots,i_t\leq l}(\mathcal{S}) = \mathbf{p}_j^{(t-1)} \circ \mathbf{p}_{j_t}^{(1)} \right\}, \tag{3.19}$$

where $\mathbf{p}_j^{(t-1)} = \mathbf{p}_{j_1}^{(1)} \circ \mathbf{p}_{j_2}^{(1)} \cdots \circ \mathbf{p}_{j_{t-1}}^{(1)} \in F_{t-1}$ and $\mathbf{p}_{j_t}^{(1)} \in F_1$. For simple expression, the candidate polynomial set is written as $C_t = \{\boldsymbol{q}_1(\mathcal{S}), \boldsymbol{q}_2(\mathcal{S}), \cdots, \boldsymbol{q}_l(\mathcal{S})\}$. Then the above orthogonal processing can be given as

$$\boldsymbol{g}_i^{(t)}(\mathcal{S}) = \boldsymbol{q}_i(\mathcal{S}) - \sum_{\mathbf{p}^{(t-1)}(\mathcal{S})\in F_t} \langle \boldsymbol{q}_i(\mathcal{S}), \mathbf{p}^{(t-1)}(\mathcal{S})\rangle \mathbf{p}^{(t-1)}(\mathcal{S}). \tag{3.20}$$

The above processing procedure performs like a consecutive processing procedure that each time one polynomial is added to the vanishing polynomial set V_t or non-vanishing polynomial set F_t. Actually, we can operate more polynomials simultaneously with singular value decomposition (SVD). Before that, let us first introduce a property similar to Theorem 2.

Theorem 3. *Let $\boldsymbol{g}^{(t)}(\mathcal{S})$ be a set of polynomials of degree t. It can be constructed as $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) = \sum_{i_1,i_2,\cdots,i_t\leq l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} \circ \cdots \circ \boldsymbol{f}_{i_t}^{(1)}$. Assume that for $i_1, i_2, \cdots, i_t \leq l$, we have that $\boldsymbol{f}_{i_1}^{(1)}, \boldsymbol{f}_{i_2}^{(1)}, \cdots, \boldsymbol{f}_{i_t}^{(1)}$ are non-vanishing on $\mathcal{S}$. Denoting $F_{t-1} = \{\mathbf{p}_1^{(t-1)}, \mathbf{p}_2^{(t-1)}, \cdots, \mathbf{p}_{|F_{t-1}|}^{(t-1)}\}$ and $F_1 = \{\mathbf{p}_1^{(1)}, \mathbf{p}_2^{(1)}, \cdots, \mathbf{p}_{|F_1|}^{(1)}\}$ as a non-vanishing polynomial set of degree $t-1$ and 1, respectively. Then any polynomials $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S})$ can be formulated as*

$$\begin{aligned} \hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) &= \sum_{i_1,i_2,\cdots,i_t\leq l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{f}_{i_2}^{(1)} \circ \cdots \circ \boldsymbol{f}_{i_t}^{(1)} \\ &= \sum_{j,j_t} \left[(\mathbf{p}_j^{(t-1)} \circ \mathbf{p}_{j_t}^{(1)}) \left(\sum_{i_t\leq l} \alpha_j^{(t-1)} \alpha_{i_t,j_t}^{(1)} \right) \right], \end{aligned} \tag{3.21}$$

where $\alpha_j^{(t-1)}$ and $\alpha_{i_t,j_t}^{(1)}$ denotes the coefficients that make $\mathbf{p}_j^{(t-1)} \circ \mathbf{p}_{j_t}^{(1)} \neq \mathbf{0}_{T\times 1}$ for all j, j_t. □

The theoretical proof is shown in the Appendix B. Then F_t can be generated from the span of $F_{t-1}\times F_1$. The matrix $\mathbf{A}_t$ can be formed as $\mathbf{A}_t = [\boldsymbol{g}_1^{(t)}(\mathcal{S}), \boldsymbol{g}_2^{(t)}(\mathcal{S}), \cdots, \boldsymbol{g}_{|F_t|}^{(t)}(\mathcal{S})]$. By using

SVD, the matrix $\mathbf{A}_t$ can be decomposed as $\mathbf{A}_t = \mathbf{L}\mathbf{D}\mathbf{U}^\top$. Using a simple matrix operation, we then obtain

$$\mathbf{A}_t\mathbf{U} = \left[\boldsymbol{g}_1^{(t)}(\mathcal{S}), \boldsymbol{g}_2^{(t)}(\mathcal{S}), \cdots, \boldsymbol{g}_{|F_t|}^{(t)}(\mathcal{S})\right]\mathbf{U} = \mathbf{L}\mathbf{D}, \tag{3.22}$$

where $\boldsymbol{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l}_T]$ and $\boldsymbol{l}_i \in \mathbb{R}^T$ for $i = 1, 2, \cdots, T$. The above equation can be written as

$$\boldsymbol{\eta}_i^{(t)}(\mathcal{S}) = \sum_{j=1}^{|F_t|} U_{j,i}\boldsymbol{g}_j^{(t)}(\mathcal{S}) = \sum_{j=1}^{T} D_{j,i}\boldsymbol{l}_j = D_{i,i}\boldsymbol{l}_i, \tag{3.23}$$

where $i = 1, 2, \cdots, |F_t|$. If $D_{i,i} < \epsilon$, we denote the polynomial $\boldsymbol{\eta}_i^{(t)}(\mathcal{S})$ vanishes, where ϵ is the tolerate value used to evaluate the polynomials how close to zero. Thus, we update $V_t \leftarrow V_t \bigcup \{\boldsymbol{\eta}_i^{(t)}(\mathcal{S})\}$. Otherwise we update $F_t \leftarrow F_t \bigcup \{\boldsymbol{\eta}_i^{(t)}(\mathcal{S})/\|\boldsymbol{\eta}_i^{(t)}(\mathcal{S})\|\}$.

3.3.2 Approximate Simultaneous Diagonalization

After we obtain a set of polynomials that projected data in the feature space, we consider the blind source separation with temporal structure employed as the separation criterion. Thus, the nonlinear separation problem can be changed to a generalized joint diagonalization problem. An alternative technique proposed in [121] can achieve the process by implementing two steps: 1. whitening and 2. Constructing several Jacobi rotations to achieve an approximate simultaneous diagonalization of the correlation matrix set. In step 1, we find a linear transform, which can be determined by taking the inverse square root of the covariance matrix as

$$\boldsymbol{\Theta}_{\boldsymbol{\phi}} = \boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}} = \left(\mathbb{E}\left[\boldsymbol{\phi}(\mathbf{x}(t))\boldsymbol{\phi}(\mathbf{x}(t))^\top\right]\right)^{-\frac{1}{2}}, \tag{3.24}$$

where $\boldsymbol{\phi}(\mathbf{x}(t)) = [g_1(\mathbf{x}(t)), g_2(\mathbf{x}(t)), \cdots, g_k(\mathbf{x}(t))]^\top$ and k is total number of vanishing polynomials. The transform $\boldsymbol{\Theta}_{\boldsymbol{\phi}}$ gives a representation of the signals $\boldsymbol{\phi}(\mathbf{x}(t))$ in a new basis and the transformed signals are denoted by $\mathbf{z}(t) = \boldsymbol{\Theta}_{\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{x}(t)) = \boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}}\boldsymbol{\phi}(\mathbf{x}(t))$. By defining a time-lagged correlation matrix of $\mathbf{z}(t)$, the form is given as

$$\begin{aligned}
\boldsymbol{\Sigma}_{\mathbf{z}(\tau)} &= \mathbb{E}\left[\mathbf{z}(t)\mathbf{z}(t+\tau)^\top\right] \\
&= \boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}}\mathbb{E}\left[\boldsymbol{\phi}(\mathbf{x}(t))\boldsymbol{\phi}(\mathbf{x}(t+\tau)^\top\right]\left(\boldsymbol{\Sigma}_{\boldsymbol{\phi}(\mathbf{x}(t))}^{-\frac{1}{2}}\right)^\top \\
&= \boldsymbol{\Theta}_{\boldsymbol{\phi}}\boldsymbol{\Sigma}_{\boldsymbol{\phi}(\tau)}\boldsymbol{\Theta}_{\boldsymbol{\phi}}^\top.
\end{aligned} \tag{3.25}$$

With different time lag, we can have different correlation matrix as $\boldsymbol{\Sigma}_{\mathbf{z}(\tau_1)}, \boldsymbol{\Sigma}_{\mathbf{z}(\tau_2)}, \cdots, \boldsymbol{\Sigma}_{\mathbf{z}(\tau_N)}$, where N is the number of time lags. After the pre-whitening step, any time delayed correlation

matrix can be transformed to a diagonal matrix by a rotation matrix $\mathbf{Q}$ as

$$\begin{cases} \mathbf{\Sigma}_{\mathbf{z}(\tau_1)} = \mathbf{Q}\mathbf{\Lambda}_{\mathbf{z}(\tau_1)}\mathbf{Q}^\top, \\ \mathbf{\Sigma}_{\mathbf{z}(\tau_2)} = \mathbf{Q}\mathbf{\Lambda}_{\mathbf{z}(\tau_2)}\mathbf{Q}^\top, \\ \qquad \vdots \\ \mathbf{\Sigma}_{\mathbf{z}(\tau_N)} = \mathbf{Q}\mathbf{\Lambda}_{\mathbf{z}(\tau_N)}\mathbf{Q}^\top. \end{cases} \tag{3.26}$$

Concatenating both the whitening matrix $\mathbf{\Theta}_{\boldsymbol{\phi}}$ and the rotation matrix $\mathbf{Q}$ yields the demixing matrix as

$$\mathbf{W} = \mathbf{Q}^{-1}\mathbf{\Theta}_{\boldsymbol{\phi}} = \mathbf{Q}^{-1}\mathbf{\Sigma}^{-\frac{1}{2}}_{\boldsymbol{\phi}(\mathbf{x}(t))}. \tag{3.27}$$

Therefore, the signal $\tilde{\mathbf{s}}(t)$ can be expressed as

$$\tilde{\mathbf{s}}(t) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}(t)) = \mathbf{Q}^{-1}\mathbf{\Sigma}^{-\frac{1}{2}}_{\boldsymbol{\phi}(\mathbf{x}(t))}\boldsymbol{\phi}(\mathbf{x}(t)). \tag{3.28}$$

Note that the dimensions of $\tilde{\mathbf{s}}(t)$ and the original source $\mathbf{s}(t)$ are k and n respectively, where $k > n$. We need to select n sources from $\tilde{\mathbf{s}}(t)$, which construct the estimation of the original sources $\mathbf{s}(t)$. Considering all the projected components, we have the demixed signals $\tilde{\mathbf{S}} = \mathbf{Q}^{-1}\mathbf{\Theta}_{\boldsymbol{\phi}}\mathbf{\Phi}$, where $\mathbf{\Phi} = [\boldsymbol{\phi}(\mathbf{x}(1)), \cdots, \boldsymbol{\phi}(\mathbf{x}(T))]$ and $\tilde{\mathbf{S}} = [\tilde{\mathbf{s}}(1), \cdots, \tilde{\mathbf{s}}(T)]$. Since the original sources are mutually independent, the demixed sources should be also independent even if the demixed matrix is applied to the signal $\tilde{\mathbf{s}}(t)$ again. Therefore, we can obtain another set of signal $\tilde{\mathbf{S}}' = [\tilde{\mathbf{s}}'(1), \tilde{\mathbf{s}}'(2), \cdots, \tilde{\mathbf{s}}'(T)]$. By employing the above temporal structure on $\tilde{\mathbf{s}}(t)$, the correlation (corr) between each row in $\tilde{\mathbf{S}}$ and each row in $\tilde{\mathbf{S}}'$ is calculated by

$$\operatorname{corr}\left(\tilde{\mathbf{s}}(t), \tilde{\mathbf{s}}'(t)\right) = \frac{\sum_{t=1}^{T}(\tilde{s}_i - \mathbb{E}[\tilde{s}_i])(\tilde{s}'_j - \mathbb{E}[\tilde{s}'_j])}{\sqrt{\sum_{t=1}^{T}(\tilde{s}_i - \mathbb{E}[\tilde{s}_i])^2}\sqrt{\sum_{t=1}^{T}(\tilde{s}'_j - \mathbb{E}[\tilde{s}'_j])^2}}. \tag{3.29}$$

Then, the rows in $\tilde{\mathbf{S}}$ with the maximum n correlations are denoted as the recovered sources $\hat{\mathbf{s}}(t)$.

3.4 Computational Complexity

3.4.1 Computational Complexity of Vanishing Polynomial

In this section, we analyze the computational complexity of the algorithm. Recalling our notations, we defined two sets: V and F are sets of vanishing polynomials and non-vanishing polynomials, respectively. The subscript of F denotes the subset of non-vanishing polynomials in the corresponding degree. For example, we use the notation $F_1 \subset F$ to denote the non-vanishing polynomials with degree 1 in $\mathbb{R}^T$. $F^{[r]} = \bigcup_{i \le r} F_i$ is defined as the union of the collection F_i up to degree r. $|F_i|$ denotes the number of polynomials in the non-vanishing polynomial set F_i. In Algorithm 1, the progress will terminate at round r when the set F_r is empty. On the other hand, the progress does not stop, then $|F^{[r]}| \ge r$ holds for any $F^{[r]} = \bigcup_{i \le r} F_i$, because F_i should contain at least one polynomial. Since $F^{[r]}$ is a set of orthonormal non-vanishing polynomials, none of the vector in $F^{[r]}$ can be expressed as the combination of other polynomials in $F^{[r]}$. Then the

Table 3.1: A Comparison of the Computational Complexity with Several Integration Methods.

TDSEP [41]	KTDSEP [42, 43, 44]	ViNLisem
$\mathcal{O}(Nn^2T)$	$\mathcal{O}(Nd^2T)$	$\mathcal{O}\left(T\sum_{i=1}^{r}\|F^{[i-1]}\|\|F_1\|\|F_{i-1}\| + N\|V\|^2T\right)$

rank of the matrix with the columns listed as the polynomials from $F^{[r]}$ is $|F^{[r]}|$. Consequently, we have $|F^{[r]}| \leq T$.

Suppose we have the non-vanishing polynomial set $F_1, F_2, \cdots, F_{r-1}$, the candidate polynomial set C_r is generated as $C_r = F_{r-1} \times F_1$. Let us enumerate all the non-vanishing polynomial according to the order in which they were inserted into $F^{[r-1]}$, which is listed as $F^{[r-1]} = \{\boldsymbol{g}_1(\mathcal{S}), \boldsymbol{g}_2(\mathcal{S}), \cdots, \boldsymbol{g}_{|F^{[r-1]}|}(\mathcal{S})\}$. Then for any polynomial from the candidate polynomial set C_r, we have

$$\boldsymbol{g}_i(\mathcal{S}) = \underbrace{\boldsymbol{\rho}_i(\mathcal{S})}_{\mathcal{O}(T)} - \underbrace{\sum_{\boldsymbol{g}(\mathcal{S})\in F^{[r-1]}} \langle \boldsymbol{\rho}_i(\mathcal{S}), \boldsymbol{g}(\mathcal{S})\rangle \boldsymbol{g}(\mathcal{S})}_{\mathcal{O}(T\times|F^{[r-1]}|)}. \tag{3.30}$$

where $\rho_i(\mathcal{S})$ is the candidate polynomial. Since $\rho_i(\mathcal{S})$ is the constant vector, it can be evaluated in time $\mathcal{O}(T)$. There are $|F^{[r-1]}|$ vector in the non-vanishing polynomial set $F^{[r-1]}$. Any polynomial $\boldsymbol{g}_i(\mathcal{S})$ can be written as a product of two polynomials from F_1 and F_{r-1} minus a linear combination of $\boldsymbol{g}_1(\mathcal{S}), \boldsymbol{g}_2(\mathcal{S}), \cdots, \boldsymbol{g}_{|F^{[r-1]}|}(\mathcal{S})$. Therefore, the process can be evaluated in time $\mathcal{O}(T \times |F^{[r-1]}|)$. A similar argument shows that if we take account into all the polynomials in the candidate polynomial set C_r, the evaluation of computational cost is $\mathcal{O}(T \times |F^{[r-1]}| \times |C_r|)$. Thus, considering the iteration up to the degree of r, it will take the computational cost as $\mathcal{O}(T\sum_{i=1}^{r}(|F^{[i-1]}||C_i|)) = \mathcal{O}(T\sum_{i=1}^{r}(|F^{[i-1]}||F_1||F_{i-1}|))$.

3.4.2 Computational Complexity of Temporal Process

Next, we consider another part of the computational cost of the temporal structure. For the observed signal $\mathbf{x}(t) \in \mathbb{R}^n$, the calculation of the covariance $\boldsymbol{\Sigma}_{\mathbf{x}}^{\frac{1}{2}} = (\frac{1}{T}\mathbf{x}\mathbf{x}^\top)^{\frac{1}{2}}$ requires $\mathcal{O}(n^2T+n^2)$. The covariance matrix with time lag τ is defined by

$$\boldsymbol{\Sigma}_{\tau(\mathbf{x})} = \mathbb{E}[\mathbf{x}(t)\mathbf{x}(t+\tau)^\top]. \tag{3.31}$$

Assume we need to calculate N time-lagged correlation matrices $\boldsymbol{\Sigma}_{\tau_1(\mathbf{x})}, \boldsymbol{\Sigma}_{\tau_2(\mathbf{x})}, \cdots, \boldsymbol{\Sigma}_{\tau_N(\mathbf{x})}$, it requires $\mathcal{O}(N(n^2T + n^2))$.

Simultaneous diagonalization of N matrices is implemented by the Jacobi-like technique [122]. We are going to search a unitary matrix that makes $\mathbf{Q}\boldsymbol{\Sigma}_{\tau_1(\mathbf{x})}\mathbf{Q}^\top, \mathbf{Q}\boldsymbol{\Sigma}_{\tau_2(\mathbf{x})}\mathbf{Q}^\top, \cdots, \mathbf{Q}\boldsymbol{\Sigma}_{\tau_N(\mathbf{x})}\mathbf{Q}^\top$ as a collection of diagonal matrices. Considering a set $\{\mathbf{Q}\boldsymbol{\Sigma}_{\tau_1(\mathbf{x})}\mathbf{Q}^\top, \mathbf{Q}\boldsymbol{\Sigma}_{\tau_2(\mathbf{x})}\mathbf{Q}^\top, \cdots, \mathbf{Q}\boldsymbol{\Sigma}_{\tau_N(\mathbf{x})}\mathbf{Q}^\top\}$ of N matrices of size $n \times n$, the process needs to take the time $\mathcal{O}(\lambda mn^2)$, where λ is the number of iterations for the simultaneous diagonalization.

After we obtain the matrix $\mathbf{Q}$, the demixing matrix is calculated as $\mathbf{W} = (\boldsymbol{\Sigma}_{\mathbf{x}}^{\frac{1}{2}}\mathbf{Q})^{-1}$, which needs the time $\mathcal{O}(2n^3 + n^2T)$. To summarize the above process, the computational cost of the

temporal process is given by

$$\mathcal{O}(n^2T + n^2 + N(n^2T + n^2) + \lambda Nn^2 + 2n^3 + n^2T). \tag{3.32}$$

Since we have $T \gg n$, $T \gg \lambda$ and $T \gg m$, the computation time of the temporal process can be approximated as $\mathcal{O}(Nn^2T)$. We have $|V|$ vectors in the vanishing polynomial set. Then the total computational cost can be evaluated in time $\mathcal{O}(T\sum_{i=1}^{r}(|F^{[i-1]}||F_1||F_{i-1}|) + N|V|^2T)$.

As shown in Table 3.1, the computational complexity of TDSEP [41] for the observed signal $\mathbf{x}(t) \in \mathbb{R}^n$ is $\mathcal{O}(Nn^2T)$, where N is the number of time lags of temporal structure. Using the approximation of multi-kernel space in [42, 43, 44], the cost of adding the signal channels from n to the high dimensional space with d that can be evaluated in $\mathcal{O}(Nd^2T)$. Since KTDSEP method sets the number of kernel spaces initially, the parameter d is fixed rather than depending on the data itself. In contrast, the algorithm ViNLisem is not restricted to any specific mixture or parameter model, but generate the multi-layer architecture to approximate such nonlinearity solely based on the data and the degree of vanishing polynomials.

3.5 Experiments with Real-World Data

In this section, experimental results of the proposed algorithms for three kinds of nonlinear mixtures are shown. The methods used for comparison and evaluation equation are presented in Section 3.5.1. Afterward, the description of data and experimental settings are shown in Section 3.5.2. The results and their performance evaluation are given in Section 3.5.3.

3.5.1 Methods and Evaluation Equation

The separation performance of the proposed nonlinear separation method is evaluated with other six approaches on five real audio datasets. The following shows the six methods used for comparison.

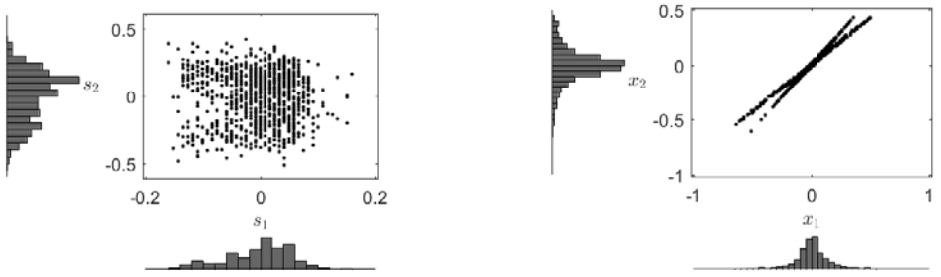
1. TDSEP [41]: Temporal decorrelation source separation relies on the estimation of simple time-lagged covariance matrices (second-order statistics), which emphasize the difference from the temporally i.i.d. case.

2. KTDSEP[3] [44]: Kernel-based TDESP was proposed by Harmeling et al. that transformed the source signals into kernel spaces. The approach relies on such kernels that are assumed to be chosen enough to approximate the nonlinearity of the observed signals.

3. FICA[4] [14]: Fast independent component analysis is a significant milestone for blind source separation. It recovered the statistically dependent sources by minimizing the criterion composed of the negative-entropy.

4. KICA[5] [32]: Kernel-based ICA is used to show the necessity of exploiting nonlinear ICA methods for separating nonlinear mixtures.

[3] http://people.kyb.tuebingen.mpg.de/harmeling/code/ktdsep-0.2.tar
[4] https://research.ics.aalto.fi/ica/fastica/
[5] http://www.di.ens.fr/ fbach/kernel-ica/index.htm



(a) Source signals (b) Mixture signals

Figure 3.2: (a) The scatter plots of the original sources use the "AMI" dataset[8] in Table 3.2. (b) The mixture signals are generated from distorted source (DS) function.

5. JADE[6] [123]: Joint approximate diagonalization of eigenmatrix is considered to operate on the high-order statistics of independence.

6. SOBI[7] [121]: Second-order blind identification is a technique to exploit the coherence of the source signals, which relies only on stationary second-order statistics.

To measure the performance of recovered sources, the normalized mean squared error (NMSE) is employed [52], which has the following definition

$$\mathrm{NMSE}(\mathbf{s}_i, \hat{\mathbf{s}}_i) = 10 \log_{10} \left(\frac{1}{n} \sum_{i=1}^{n} \min_{\delta} \frac{\|\mathbf{s}_i - \delta \hat{\mathbf{s}}_i\|_2^2}{\|\mathbf{s}_i\|_2^2} \right), \tag{3.33}$$

where $\hat{\mathbf{s}}_i$ denotes the estimate of the source signal $\mathbf{s}_i$, and δ is a scalar reflecting the scalar ambiguity.

3.5.2 Data and Experiment Setting

The experiments are designed on the assumption that the observed signals are mixed nonlinearly. The sources used for the following simulations include 5 real-world audio signals with different temporal properties. They are publicly available [1]. Each one has its own advantages, depending on whether one is interested in a variety of environments, in a number of microphones, or in the overlap. For instance, the data "AMI" has two kinds of sound from the cable news and network news. Another data "Multitrack" was mixed with two anonymous singers. All the sources were sampled at 8,000 Hz. The length of the samples was varied to assess how the amount of training data affects the performance of the algorithm. The general properties of the datasets are summarized in Table 3.2.

Three kinds of nonlinear mixture functions were investigated, including the distorted source (DS) in [104], the post-nonlinear mixture (PNL) in [33], and the generic nonlinear (GN) in [124, 31].

The distorted source (DS) In the DS mixture function of (3.34), each observation is a linear mixture of nonlinear distorted sources. Specifically, in the experiments the two channel mixtures

[6]http://perso.telecom-paristech.fr/ cardoso/Algo/Jade/jadeR.m
[7]https://github.com/aludnam/MATLAB/blob/master/sobi/sobi.m

Table 3.2: Descriptions of Real-World Data [1].

Name	Scenario	Duration(s)	Microphones	Overlap
AMI[a]	News	100	16	yes
CHiME3[b]	Talker	19	6	yes
Nonspeech[c]	Wind	20	4	no
SiSEC[d]	TV order	6	16	no
Multitrack[e]	Theater	38	20	yes

[a] https://research.ics.aalto.fi/ica/newindex.shtml;
[b] http://laslab.org/SpeechSeparationChallenge/;
[c] http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html;
[d] http://sisec2010.wiki.irisa.fr/tiki-index.html;
[e] http://www.cambridge-mt.com/ms-mtk.htm

were generated according to

$$\begin{aligned} x_1(t) &= a_1 s_1(t) + 3\tanh(s_2(t)/4) + 0.1 s_2(t), \\ x_2(t) &= a_2 s_2(t) + 3\tanh(s_1(t)/4) + 0.1 s_1(t), \end{aligned} \tag{3.34}$$

where $a_1 = a_2 = 1$. Fig. 3.2 (a) shows the scatter plot of the sources $\mathbf{s}_i(t)$ and that of the observations $\mathbf{x}_i(t)$. To see the level of nonlinear distortion in the mixing transformation, we give the scatter plot of the affine transformation of $\mathbf{s}_i(t)$ in Fig. 3.2 (b) .

The post-nonlinear (PNL) The post-nonlinear mixtures constitute a particularly interesting example of the theoretical separability characterized by weak indeterminacy. The sources were the first subject to a linear mixture $\mathbf{z}(t) = \mathbf{A}\mathbf{s}(t)$, where $\mathbf{A}$ is a 2×2 mixing matrix give by

$$\mathbf{A} = \begin{pmatrix} -0.2261 & -0.1189 \\ -0.1706 & -0.2836 \end{pmatrix}. \tag{3.35}$$

Then each mixture component is generated from a nonlinear, invertible transformation, as the form of

$$\begin{aligned} x_1(t) &= (z_2(t) + 3z_1(t) + 6)\cos(1.5\pi) z_1(t), \\ x_2(t) &= (z_2(t) + 3z_1(t) + 6)\sin(1.5\pi) z_1(t). \end{aligned} \tag{3.36}$$

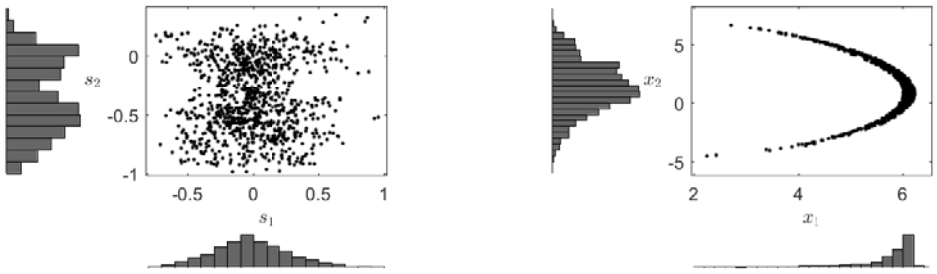
The sources are plotted in Fig. 3.3 (a). The mixture components are shown in Fig. 3.3 (b), where we can see the distortions caused by the nonlinearities.

The generic nonlinear (GN) In the following example, at each sample t, the sources are mixed nonlinearly as

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \cos\alpha(s(t)) & -\sin\alpha(s(t)) \\ \sin\alpha(s(t)) & \cos\alpha(s(t)) \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix}, \tag{3.37}$$

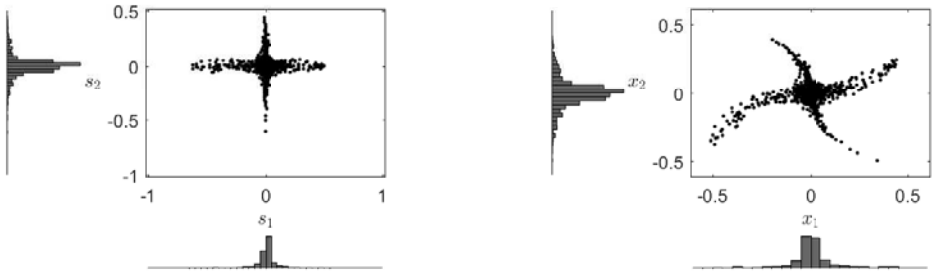
where $\alpha(s(t))$ is defined by the parameter model

$$\alpha(s(t)) = \alpha_0 + \gamma \times \sqrt{s_1^2(t) + s_2^2(t)}.$$



(a) Source signals (b) Mixture signals

Figure 3.3: (a) The scatter plots of the original sources use the "ChiME3" dataset[9] in Table 3.2. (b) The mixture signals are generated from post-nonlinear (PNL) function.



(a) Source signals (b) Mixture signals

Figure 3.4: (a) The scatter plots of the original sources use the "Nonspeech" dataset[10] in Table 3.2. (b) The mixture signals are generated from generic nonlinear (GN) function.

In our simulation, the parameter α_0 and γ are set to 0 and 1, respectively.

Fig. 3.4 (a) illustrates the source signals, which is the case for the audio data of "Nonspeech" collected in Table 3.2. By using a mixing function given in (3.37), the observations are nonlinearly mixed, which is shown as an anchor-shaped structure in Fig. 3.4 (b). The mixing function (3.37) is not symmetric in $\mathbf{s}_1(t)$ and $\mathbf{s}_2(t)$. Thus, for every pair of sources, there are two possible mixtures and we have tested both for each source pair.

For most blind source separation method based on the temporal structure, such as TDSEP, KTDSEP and our proposed ViNLisem method, the selection of the optimal time lags is a tough problem. Clearly, the performance can be degraded if the improper delay is chosen, whereas a large number of delays always give a stable solution. Here, we got some knowledge of practical experiments, which was shown in Fig. 3.5 that many delays always brings us to the stable side. Thus, in the following experiments, the time-shift is set as $\tau_{\mathrm{TDSEP}} = 0, 1, \cdots, 20$, $\tau_{\mathrm{KTDSEP}} = 0, 1, \cdots, 40$ and $\tau_{\mathrm{ViNLisem}} = 0, 1, \cdots, 7$, respectively.

In addition, for the best parameter setting, we could apply KTDSEP with a polynomial kernel of degree 9, i.e. $\mathcal{K}(\mathbf{s}_1, \mathbf{s}_2) = (\mathbf{s}_1^\top \mathbf{s}_2 + 1)^9$ and the dimensionality of kernel space set as 20. In practice, the real data are noisy that allow us to consider a tolerate value ϵ, so as to the polynomials almost vanish, i.e. $\|g_i(\mathbf{x})\| \leq \epsilon$. The parameter ϵ is used to indicate the distance between the measured polynomials and the value 0. If a bigger ϵ is selected, the polynomials will have a bigger distance from the value 0. However, if a smaller ϵ is selected, the degree of the polynomial will be higher to make the polynomial satisfying the restrict of ϵ. Then the cost time will be longer to search such polynomial. Therefore, we set the parameter $\epsilon = 0.001$ according to the experiments of the real datasets. The additive noise is generated to be white and Gaussian with

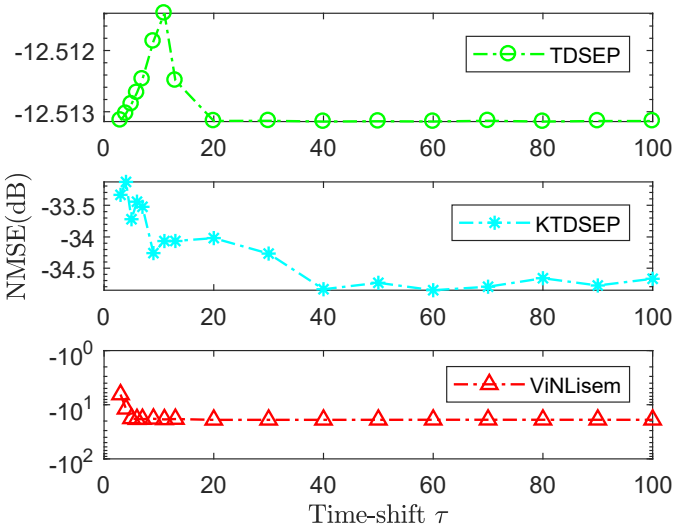


Figure 3.5: The Performance Indexes on Various Time Shift τ. The methods with temporal structure are TDSEP, KTDSEP, and our proposed ViNLisem, respectively.

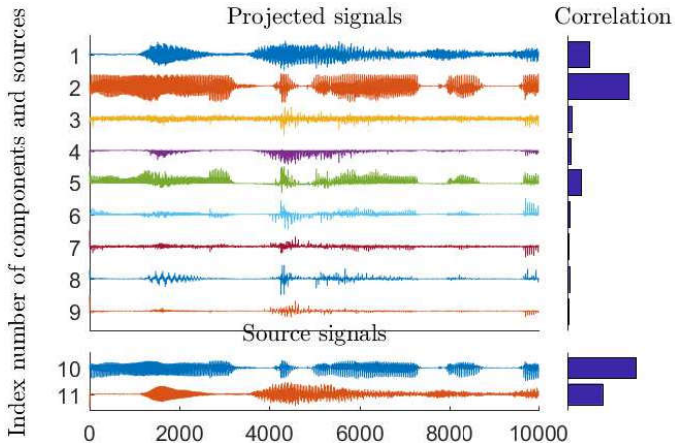
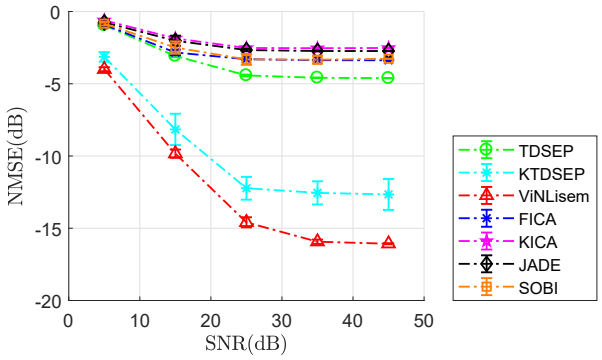


Figure 3.6: All the Projected Components and the Original Sources. The horizontal bars indicate the normalized correlation.

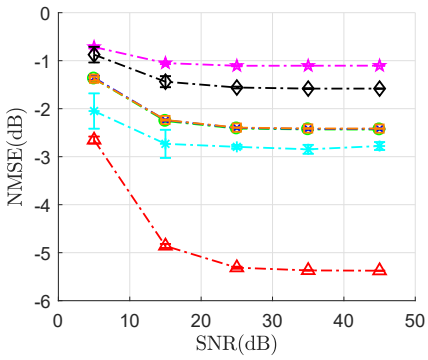
uncorrelated samples whose variance was assumed to be uniform. The algorithms are performed under the signal-to-noise power ratio (SNR) varied from 5 dB to 45 dB by a step of 10 dB. To reduce the randomness effect, 20 times of Monte Carlo simulations are performed to evaluate the performance of the algorithms versus different SNR.

3.5.3 Results

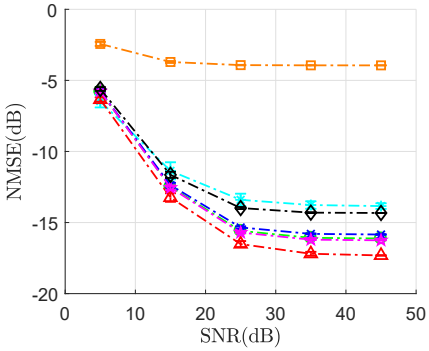
Since our algorithm utilizes a set of polynomials to approximate the nonlinearity of mixture, we thus obtain 9 components (projected signals) adaptively for dataset "AMI" as shown in Fig. 3.6. Then, two components with the maximum correlation are selected as described in the previous section. The best matching waveforms with the maximum correlations are shown as the first and second rows, which are denoted as the estimation of original signals $\hat{\mathrm{s}}_1$ and $\hat{\mathrm{s}}_2$, respectively. The



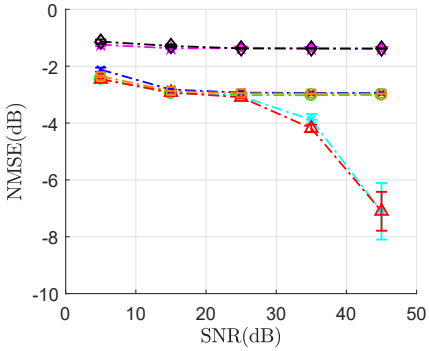
(a) The accuracy for DS mixture on Data 1.



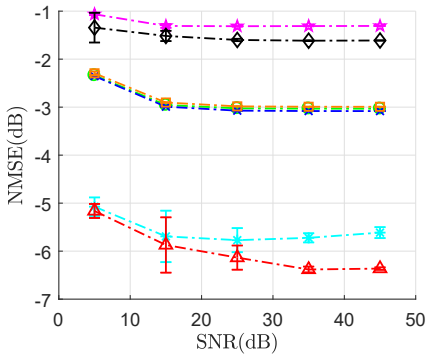
(b) The accuracy for PNL mixture on Data 2.



(c) The accuracy for GN mixture on Data 3.



(d) The accuracy for GN mixture on Data 4.



(e) The accuracy for GN mixture on Data 5.

Figure 3.7: The separation performance comparison for three kinds of mixed functions in which the different dataset in Table 3.2 are used.

algorithm automatically chooses two signals that turn out to reach very high correlation coefficients (cc), such as $cc(\mathbf{s}_1, \hat{\mathbf{s}}_1) = 0.9848$ and $cc(\mathbf{s}_2, \hat{\mathbf{s}}_2) = 0.9803$.

To clarify the separation performance, we use the NMSE in (3.33) as the error measure. We evaluate seven BSS approaches on three kinds of mixed functions with five different datasets. Fig. 3.7 show parts of the experimental results. Similar accuracy trends were also observed with other datasets being used to testify different mixed functions with different BSS approaches. We can see from Fig. 3.7 that the ViNLisem achieved a more accurate estimate than the other methods. In contrast, FICA and KICA optimized their estimate by having access to all the samples in one space. In addition, we also verified that for all datasets, the improved performance of the proposed approach was significant. Apart from the estimation quality, an important aspect for ViNLisem

method is that the vanishing components are constructed solely on the input data without any additional constraints on the mixing functions except for invertibility.

Among these methods used for comparison, we can distinguish two classes. Methods such as JADE and Fast ICA are based on statistics of order higher than two, which require at most one source can be Gaussian. This means that their performance will be poor if more than one source is close to Gaussian. However, in practice, most of the sources have distributions deviate markedly from Gaussian (e.g. speech data are strongly super-Gaussian, while images tend to be strongly sub-Gaussian). Methods of this class do not exploit any temporal or spatial structure of the sources. On the other hand, methods such as TDSEP and SOBI use only second-order statistics, and can deal with any number of Gaussian sources. However, they require sources being with temporal structure. Again, most sources of practical interest (such as speech, biomedical signals or images) do not have a temporal or spatial structure that can be used.

Note that unlike KTDSEP, ViNLisem does not assume the number of approximate functions initially, but adapt to the nonlinear approximation in the form of a multi-layer representation. Therefore, the complexity and storage requirements of the model are proportional to the number of vanishing components. The complexity of the models learned by ViNLisem is generally larger than that of the KTDSEP.

3.6 Conclusion

Our work has three main contributions. First, the approach presents a novel mathematical construction with a multi-layer architecture. By using the layer-by-layer representation, we can approximate such nonlinearity of mixing functions. Similar to the principle of modern deep learning, the layers are generated one-by-one up to the higher-degree representations of data. Once such representations are generated, a final output layer is constructed by solving a convex optimization problem. Thus, the technique establishes a highly useful isomorphism between the projection of the data points and the multi-layer representations. By projecting a time-invariant nonlinear BSS to the local linear problem, the nonlinear problem can be linearly separable. Importantly, the parameters and forms of polynomials depend solely on the input data, which guarantees the robustness of the structures. We thus address the general problem without being restricted to any specific mixture or parametric model.

Then, the layer-by-layer representation is adaptively generated solely on the observations. As the number of spanned spaces goes up, the computational complexity grows exponentially. To overcome this obstacle, relying on the properties of vanishing polynomials, we provide a feasible way to reduce the computational cost as shown in Theorem 2 and Theorem 3. Finally, considering the temporal correlation as the separation criterion, the approach can be designed by emphasizing the difference from the temporally i.i.d. data. Therefore, we can break the nonlinear problem down into a simpler version of the generalized joint diagonalization problem in the feature space. However, due to adopting the nonlinear approximation in the form of a sample representation, the complexity and storage requirements of the model are proportional to the number of vanishing components, which is generally larger than that of the TDSEP and KTDSEP.

3.7 Appendix

3.7.1 Proof of Theorem 2

For instance, considering the polynomials of degree 2, we set $\rho_{i_1,i_2}(\mathbf{x}(t)) = x_{i_1}(t)x_{i_2}(t)$, for all i_1 and i_2. Thus, we now need to consider n^2+n+1 columns. As the degree goes up, the number of columns increases exponentially. To overcome this obstacle, we propose a method to reduce the computational cost relying on the underlying structure and the property of the vanishing ideal

Proof. Denoting $F_1 = \{\mathbf{p}_1^{(1)}, \mathbf{p}_2^{(1)}, \cdots, \mathbf{p}_{|F_1|}^{(1)}\}$ as a non-vanishing polynomial set of degree 1, where $|F_1|$ denotes the number of elements included in the set F_1. Any polynomial of degree 1 generated from F_1 can be expressed as

$$\boldsymbol{f}_{i_1}^{(1)} = \sum_{j_1} \alpha_{i_1,j_1}^{(1)} \mathbf{p}_{j_1}^{(1)}, \quad \boldsymbol{h}_{i_2}^{(1)} = \sum_{j_2} \alpha_{i_2,j_2}^{(1)} \mathbf{p}_{j_2}^{(1)}, \tag{3.38}$$

where $\alpha_{i_1,j_1}^{(1)}$ and $\alpha_{i_2,j_2}^{(1)}$ denote the coefficients that make $\boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{h}_{i_2}^{(1)} \neq \mathbf{0}_{T\times 1}$ for all $i_1, i_2 \leq l$. Then F_2 can be generated from the span of $\boldsymbol{f}_{i_1}^{(1)}$ and $\boldsymbol{h}_{i_2}^{(1)}$ for $i_1, i_2 \leq l$ as

$$\begin{aligned}
\hat{\boldsymbol{g}}^{(2)}(\mathcal{S}) &= \sum_{i_1,i_2\leq l} \boldsymbol{f}_{i_1}^{(1)} \circ \boldsymbol{h}_{i_2}^{(1)} \\
&= \sum_{i_1,i_2\leq l} \Big(\sum_{j_1} \alpha_{i_1,j_1}^{(1)} \mathbf{p}_{j_1}^{(1)} \Big) \Big(\sum_{j_2} \alpha_{i_2,j_2}^{(1)} \mathbf{p}_{j_2}^{(1)} \Big) \\
&= \sum_{j_1,j_2} \Big[(\mathbf{p}_{j_1}^{(1)} \circ \mathbf{p}_{j_2}^{(1)}) \Big(\sum_{i_1,i_2\leq l} \alpha_{i_1,j_1}^{(1)} \alpha_{i_2,j_2}^{(1)} \Big) \Big],
\end{aligned} \tag{3.39}$$

where the polynomials are assumed to be composed of linear functions that each linear function is described by a coefficient vector $\boldsymbol{\alpha} \in \mathbb{R}^{n+1}$. And $\alpha_{i_1,j_1}^{(1)}$ is the coefficient that corresponds to the i_1-th element of the candidate set C_1, which is used to weight the j_1-th element $\mathbf{p}_{j_1}^{(1)}$ of the non-vanishing polynomial set F_1. Thus, we have $\hat{\boldsymbol{g}}^{(2)}(\mathcal{S})$ generated from the span of $F_1 \times F_1$ and that can be used to construct F_2 and V_2. □

3.7.2 Proof of Theorem 3

Constructing the Polynomials of Degree 3

Considering the polynomials of degree 3, we set $\rho_{i_1,i_2,i_3}(\mathbf{x}(t)) = x_{i_1}(t)x_{i_2}(t)x_{i_3}(t)$, for all i_1, i_2 and i_3. Then $\hat{\boldsymbol{g}}^{(3)}(\mathcal{S})$ is generated from the span of $F_1 \times F_2$.

Proof. Denoting $F_2 = \{\mathbf{p}_1^{(2)}, \mathbf{p}_2^{(2)}, \cdots, \mathbf{p}_{|F_2|}^{(2)}\}$ as a non-vanishing polynomial set of degree 2, where $|F_2|$ denotes the number of elements included in the set F_2. Similarly, any polynomial of degree 3 can be expressed as

$$\boldsymbol{g}^{(3)}(\mathcal{S}) = \sum_{i_1,i_2,i_3} \alpha_{i_1,i_2,i_3} \boldsymbol{\rho}_{i_1,i_2,i_3}(\mathcal{S}). \tag{3.40}$$

The polynomial $\hat{\boldsymbol{g}}^{(3)}(\mathcal{S}) = \sum_{i_1,i_2,i_3 \le l} \boldsymbol{\rho}_{i_1,i_2,i_3}$ satisfies $\hat{\boldsymbol{g}}^{(3)}(\mathcal{S}) = \boldsymbol{g}^{(3)}(\mathcal{S})$ for $i_1, i_2, i_3 \le l$ for assumption. Then, $\boldsymbol{g}^{(3)}(\mathcal{S})$ can be approximated as

$$\begin{aligned}\hat{\boldsymbol{g}}^{(3)}(\mathcal{S}) &= \sum_{i_1,i_2,i_3} \alpha_{i_1,i_2,i_3}\boldsymbol{\rho}_{i_1,i_2,i_3} \\ &= \sum_{i_1,i_2,i_3} \Big(\sum_{j_1} \alpha^{(1)}_{i_1,j_1}\mathbf{p}^{(1)}_{j_1}\Big)\Big(\sum_{j_2} \alpha^{(1)}_{i_2,j_2}\mathbf{p}^{(1)}_{j_2}\Big)\Big(\sum_{j_3} \alpha^{(1)}_{i_3,j_3}\mathbf{p}^{(1)}_{j_3}\Big) \\ &= \sum_{i_1,i_2,i_3} \Big(\sum_{j_1,j_2} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\Big)\Big(\sum_{j_3} \alpha^{(1)}_{i_3,j_3}\mathbf{p}^{(1)}_{j_3}\Big). \end{aligned} \tag{3.41}$$

Since $\sum_{j_1,j_2} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}$ is in the span of $F_1 \times F_1$, thus it can be expressed as

$$\sum_{j_1,j_2} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2} = \sum_j \alpha^{(2)}_j \mathbf{p}^{(2)}_j. \tag{3.42}$$

Then (3.41) can be written as

$$\begin{aligned}\hat{\boldsymbol{g}}^{(3)}(\mathcal{S}) &= \sum_{i_1,i_2,i_3 \le l} \Big(\sum_{j_1,j_2} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\Big)\Big(\sum_{j_3} \alpha^{(1)}_{i_3,j_3}\mathbf{p}^{(1)}_{j_3}\Big) \\ &= \sum_{i_3 \le l} \Big(\sum_j \alpha^{(2)}_j \mathbf{p}^{(2)}_j\Big)\Big(\sum_{j_3} \alpha^{(1)}_{i_3,j_3}\mathbf{p}^{(1)}_{j_3}\Big) \\ &= \sum_{j,j_3} \mathbf{p}^{(2)}_j \mathbf{p}^{(1)}_{j_3}\Big(\sum_{i_3 \le l} \alpha^{(2)}_j \alpha^{(1)}_{i_3,j_3}\Big). \end{aligned} \tag{3.43}$$

Thus, $\hat{\boldsymbol{g}}^{(3)}(\mathcal{S})$ is generated from the span of $F_2 \times F_1$ that can be used to construct F_3 and V_3.

Constructing the Polynomials of Higher Degree

Similar to the above processing procedure, any polynomial of degree t can be expressed as

$$\boldsymbol{g}^{(t)}(\mathcal{S}) = \sum_{i_1,i_2,\cdots,i_t} \alpha_{i_1,i_2,\cdots,i_t}\boldsymbol{\rho}_{i_1,i_2,\cdots,i_t}(\mathcal{S}). \tag{3.44}$$

The polynomial $\boldsymbol{g}^{(t)}(\mathcal{S}) = \sum_{i_1,i_2,\cdots,i_t} \boldsymbol{\rho}_{i_1,i_2,\cdots,i_t}$ satisfies $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) = \boldsymbol{g}^{(t)}(\mathcal{S})$ for $i_1, i_2, \cdots, i_t \le l$. Denoting $F_{t-1} = \{\mathbf{p}^{(t-1)}_1, \cdots, \mathbf{p}^{(t-1)}_{|F_{t-1}|}\}$, $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S})$ can be written as (3.45).

$$\begin{aligned}\hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) &= \sum_{i_1,i_2,\cdots,i_t \le l} \boldsymbol{\rho}_{i_1,i_2,\cdots,i_t} \\ &= \sum_{i_1,i_2,\cdots,i_t \le l} \Big(\sum_{j_1} \alpha^{(1)}_{i_1,j_1}\mathbf{p}^{(1)}_{j_1}\Big)\Big(\sum_{j_2} \alpha^{(1)}_{i_2,j_2}\mathbf{p}^{(1)}_{j_2}\Big)\cdots\Big(\sum_{j_t} \alpha^{(1)}_{i_t,j_t}\mathbf{p}^{(1)}_{j_t}\Big) \\ &= \sum_{i_1,i_2,\cdots,i_t \le l} \Big(\sum_{j_1,j_2,\cdots,j_{t-1}} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2}\cdots\alpha^{(1)}_{i_{t-1},j_{t-1}})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\cdots\mathbf{p}^{(1)}_{j_{t-1}}\Big)\Big(\sum_{j_t} \alpha^{(1)}_{i_t,j_t}\mathbf{p}^{(1)}_{j_t}\Big) \end{aligned} \tag{3.45}$$

Since $\sum_{j_1,j_2,\cdots,j_{t-1}} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2}\cdots\alpha^{(1)}_{i_{t-1},j_{t-1}})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\cdots\mathbf{p}^{(1)}_{j_{t-1}}$ is in the span of $F_{t-2}\times F_1$, thus it can be expressed as

$$\sum_{j_1,j_2,\cdots,j_{t-1}} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2}\cdots\alpha^{(1)}_{i_{t-1},j_{t-1}})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\cdots\mathbf{p}^{(1)}_{j_{t-1}} = \sum_j \alpha^{(t-1)}_j \mathbf{p}^{(t-1)}_j. \tag{3.46}$$

Then (3.45) can be rewritten as

$$\begin{aligned}
\hat{\boldsymbol{g}}^{(t)}(\mathcal{S}) &= \sum_{i_1,i_2,\cdots,i_t\le l} \left(\sum_{j_1,j_2,\cdots,j_{t-1}} (\alpha^{(1)}_{i_1,j_1}\alpha^{(1)}_{i_2,j_2}\cdots\alpha^{(1)}_{i_{t-1},j_{t-1}})\mathbf{p}^{(1)}_{j_1}\mathbf{p}^{(1)}_{j_2}\cdots\mathbf{p}^{(1)}_{j_{t-1}} \right) \left(\sum_{j_t} \alpha^{(1)}_{i_t,j_t}\mathbf{p}^{(1)}_{j_t} \right) \\
&= \sum_{i_t\le l} \left(\sum_j \alpha^{(t-1)}_j \mathbf{p}^{(t-1)}_j \right) \left(\sum_{j_t} \alpha^{(1)}_{i_t,j_t}\mathbf{p}^{(1)}_{j_t} \right) \\
&= \sum_{j,j_t} \mathbf{p}^{(t-1)}_j \mathbf{p}^{(1)}_{j_t} \left(\sum_{i_t\le l} \alpha^{(t-1)}_j \alpha^{(1)}_{i_t,j_t} \right),
\end{aligned} \tag{3.47}$$

where the polynomials are assumed to be composed of linear functions that each linear function is described by a coefficient vector $\boldsymbol{\alpha}\in\mathbb{R}^{n+1}$, and $\alpha^{(1)}_{i_t,j_t}$ is the coefficient that corresponds to the i_t-th element of the candidate set C_1, and that is used to weight the j_1-th element $\mathbf{p}^{(1)}_{j_1}$ of the non-vanishing polynomial set F_1. Therefore, we can generate $\hat{\boldsymbol{g}}^{(t)}(\mathcal{S})$ only in the span of $F_{t-1}\times F_1$ rather than considering all the extension space. □

Chapter 4

A Closed-Form Expression for Nonlinear Approximation

Chapter 2 introduces a novel model based on the multi-subspace representation to extract the nonlinearity of mixing function. In practice, the approximation function is derived from some estimation algorithm with a finite sample size that even larger estimation error appears with improper model construction. In Chapter 3, we work on the convergence and asymptotic analysis of the separation approach, where the nonlinearity of the mixture function is extracted by the flexible approximation and the nonlinear problem is solved linearly in the feature space. In Chapter 4.2, we introduce the separation model that is referred to as ViNLisem algorithm. Then, the problem formulation is given mathematically. Chapter 4.3 provides a novel EM algorithm to estimate the coefficient matrix by an online recursive version. Then, a closed-form expression is used for bounding the covariance matrix presented in Chapter 4.4. Numerical experiments are carried out to corroborate the theoretical results in Chapter 4.5. We conclude the results in Chapter 4.6.

4.1 Introduction

The purpose of independent component analysis (ICA) and blind source separation (BSS) [15, 14, 125], is to extract m mutually independent elements from n observed mixtures. Consider the following linear instantaneous mixing system with m inputs and n outputs as

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \quad t = 1, 2, \cdots, T, \tag{4.1}$$

where $\mathbf{s}(t) = [s_1(t), s_2(t), \cdots, s_m(t)]^\top$ are the signals with m channels, $s_i(t)$ denotes the sample of the i-th source at time index t. The superscript $[\cdot]^\top$ denotes the transpose operation. $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^\top$ denotes the observed mixtures with n channels, which is assumed to be generated by a $n \times m$ mixing matrix $\mathbf{A}$ and the source signals $\mathbf{s}(t)$.

Commonly, the separation process of ICA is conducted on the assumption that the sources vectors are statistically independent [20]. For a linear mixing model, if the number of sources equals that of channels ($m = n$), the demixng matrix $\mathbf{W}$ can be defined as $\mathbf{W} = \mathbf{A}^{-1}$. The recovered signals are represented as $\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t)$. The linear BSS aims at estimating $\mathbf{W}$ and recovered signals $\hat{\mathbf{s}}(t)$ using only the observed signals $\mathbf{x}(t)$.

An obvious extension for the task of BSS is that the observed signals are assumed to be generated from a set of sources by a nonlinear, instantaneous and invertible function $\mathcal{F}$, i.e., $\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t))$ for all $t = 1, 2, \cdots, T$. Roughly, the nonlinear blind source separation seeks to find the mixing function (or its inverse function $\mathcal{G} = \mathcal{F}^{-1}$), solely based on the assumption that the sources are statistically independent. However, the indeterminacies imposed by the nonlinear model are difficult to handle [23, 24]. The obstacle for the nonlinear BSS problem is that solutions are non-unique without extra constraints [29]. The recovery inconsistency has been tackled by adding further prior information directly to the model or as a regularization term in the optimization processing procedure.

Most nonlinear algorithms utilize single approximation to extract the nonlinearity, such as multi-layer perceptron (MLP) in neural network [27, 30], which is employed for estimating the nonlinear separation function. By restricting the smoothness of the target transforming, MLP provides the regularized solutions to ensure that nonlinear ICA leads to the sources separable. However, the example presented in [31] shows that the smoothness property is not a sufficient

condition for this purpose. Hyvärinen and Pajunen [29] show a conformal mapping may helpful. Nonlinear ICA [32] is able to estimate a separation mapping up to the rotation when the mapping functions are restricted to the set of conformal mapping. Unfortunately, the angle preservation conditions seem very restrictive [33]. In particular, it is not realistic in the framework of the nonlinear mappings associated with the nonlinear sensor array.

A novel approach named as Vanishing Ideal based NonLinear SEparation Model (ViNLisem) was proposed in [126], which relies on a novel mathematical construction with multi-layer architecture. By considering the situation where a set of flexible approximations are utilized to extract the nonlinearity, the approach breaks the nonlinear distortion down into the version of the linear case in the feature space.

Nevertheless, the approximation function is generated adaptively depending solely on the input data. Then the true model could be different from its empirical counterpart that is assumed to be derived by the estimation algorithm with the finite sample size, which is called to be mismatched or misspecified [47]. The real data often exposes the limitations of any assumed model, since modeling errors at some level are always presented. Therefore, understanding the possible performance loss to the model misspecification is of practical interest and critical. In this chapter, we work on the convergence and asymptotic analysis of an approximation function, so as to propose a novel algebraic formalization as well as derive an upper bound on the estimation error.

This chapter provides a theoretical analysis to ViNLisem algorithm [126], which forms the closed-form expressions on the mean squared error (MSE), as well as proposing a new algebraic formalization that leads to the upper bound on the performance error. The analysis stems from the performance of a mismatched estimator that accesses the finite sample size, which is explored by two parts. One is to derive an iterative expression from the perspective of the expectation-maximization (EM) algorithm [127]. Another one is to establish the closed-form expression for bounding the covariance matrix under both the operator norm and a class of tapering estimators [128].

We proposed a novel EM algorithm to estimate the coefficient matrix $\mathbf{W}$, which is modeled as deterministic but depends on the dataset. To estimate the hidden variable, the E-step is used to obtain a convergence point of the maximum likelihood estimator, which could be interpreted as the stationary point that minimizes the Kullback-Leibler (KL) divergence. In the M-step, the hidden parameter is used to update the coefficient matrix by an online recursive approach. Then, we establish a closed-form expression for bounding the covariance matrix, as well as measuring the mis-specification problems of non-parametric function with the finite sample size.

4.2 Model and Problem Formulation

The nonlinear BSS problem is formally described as follows. The observed mixture $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^\top$ is assumed to be generated from a set of statistically independent sources $\mathbf{s}(t) = [s_1(t), s_2(t), \cdots, s_n(t)]^\top$ by a nonlinear, instantaneous and invertible function as

$$\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t)), \quad t = 1, 2, \cdots, T, \tag{4.2}$$

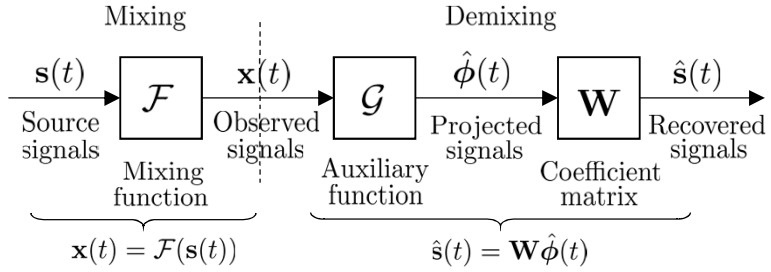


Figure 4.1: A Graphical Model for the Proposed Nonlinear BSS. The block $\mathcal{F}$ are generic nonlinear functions that lead to a mixture process. The observed signals are $\mathbf{x}(t)$, which are assumed to be generated from source signals by a nonlinear mixing function. The $\mathcal{G}$ block in the demixing process, implementing a flexible approximation, as the auxiliary function is used to match the nonlinearity of mixing functions. Thus, the projected signals $\hat{\boldsymbol{\phi}}(t)$ can make the problem linearly separable. The block $\mathbf{W}$ is a coefficient matrix, performing a linear operator that derive the estimator of original signals from the projected signals.

where t is the sample (time) index. This process can be described on the left-hand side of Fig. 4.1, which is denoted as a mixing system.

However, without any extra constraints for the mixing function, the solutions are non-unique [29]. The approach in [126] was proposed to tackle the ill-posedness with a few assumptions. By utilizing a flexible approximation to match the nonlinearity, the distortion of mixing functions can be transformed into the version of the linear case in the feature space. This process described on the right-hand side of Fig. 4.1, which is denoted as the demixing system.

Given a set of auxiliary functions that allowed us to construct the nonlinear variants by some vanishing polynomials, such as $g_1(\mathbf{x}(t)), g_2(\mathbf{x}(t)), \cdots, g_k(\mathbf{x}(t)) \in \mathcal{G}$, where $g_i(\mathbf{x}(t))$ is i-th vanishing polynomial that the observed signals $\mathbf{x}(t)$ are mapped implicitly into the feature space $\boldsymbol{\Phi} : \mathbb{R}^n \to \mathbb{R}^k$, i.e., $\hat{\phi}_i(\mathbf{x}(t))$ represents the projected value from polynomial $\hat{\phi}_i(\mathbf{x}(t)) = g_i(\mathbf{x}(t))$. The feature space is spanned from such polynomial functions that enable us to work on $\mathcal{G}$. Thus, the projected data in the feature space lead to a linear combination on the demixing process

$$\hat{\mathbf{s}}_j(t) = \sum_i W_{ji}\hat{\phi}_i(\mathbf{x}(t)), \tag{4.3}$$

where W_{ji} denotes the element of j-th row i-th column in the coefficient matrix $\mathbf{W}$. $\hat{\phi}_i(\mathbf{x}(t))$ is the projected signals that are assumed to be derived as the estimation with the finite samples size in the feature space, denoted as $\hat{\phi}_i(t)$ for short.

In this chapter, we work on a theoretical analysis of the proposed separation model [126] as described in Fig. 4.1, so that to measure the accuracy of the recovered signals. In other words, the problem consists in estimating $\hat{\mathbf{s}}(t)$ to give a closed-form expression for the mean squared error (MSE), as well as proposing a new algebraic formalization that leads to the upper bound.

First, the mean squared error (MSE) is used to measure the accuracy of the recovered signals $\hat{\mathbf{s}}(t)$ as

$$\widehat{\mathrm{MSE}} \triangleq \frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{s}}(t)-\mathbf{s}(t)\|_F^2 = \frac{1}{T}\sum_{t=1}^{T}\left\|\mathbf{W}\hat{\boldsymbol{\phi}}(t)-\mathbf{W}\boldsymbol{\phi}(t)\right\|_F^2, \tag{4.4}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm. Since we assume that the coefficient matrix $\mathbf{W}$ is fixed, which only depends on the observed signals. Thus, the source signals $\mathbf{s}(t)$ is a linear combination of $\boldsymbol{\phi}(t)$ and $\mathbf{W}$.

The theoretical analysis only considers the discrepancy between $\boldsymbol{\phi}(t)$ and its counterpart that is assumed to be derived by the estimation algorithm on the finite sample size. We thus begin by defining a convenient error term. In order to focus on well-defined accuracy, we consider the error by

$$\boldsymbol{\delta}_\phi(t) = \hat{\boldsymbol{\phi}}(t) - \boldsymbol{\phi}(t), \tag{4.5}$$

where $\boldsymbol{\phi}(t)$ is a true projected signal and $\hat{\boldsymbol{\phi}}(t)$ is its counterpart with the finite sample size. Thus MSE can be rewritten as

$$\begin{aligned}\widehat{\mathrm{MSE}} &= \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{W}\boldsymbol{\delta}_\phi(t)\|_F^2 = \frac{1}{T}\sum_{t=1}^{T}\mathrm{tr}\left\{\mathbf{W}\boldsymbol{\delta}_\phi(t)\boldsymbol{\delta}_\phi(t)^\top\mathbf{W}^\top\right\}\\ &= \mathrm{tr}\left\{\mathbf{W}\bar{\boldsymbol{\Sigma}}_{\delta_\phi}\mathbf{W}^\top\right\}.\end{aligned} \tag{4.6}$$

The second equality used the definition of Frobenius norm, i.e., $\|\mathbf{A}\|_F^2 = \mathrm{tr}\{\mathbf{A}\mathbf{A}^\top\}$. In the third equality, the results are derived from the definition of empirical counterpart $\bar{\boldsymbol{\Sigma}}_{\delta_\phi} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\delta}_\phi(t)\boldsymbol{\delta}_\phi(t)^\top$. For the true projected signals $\boldsymbol{\phi}(t)$, the linear model (4.3) implies the relation that

$$\boldsymbol{\Sigma}_s = \mathbf{W}\boldsymbol{\Sigma}_\phi\mathbf{W}^\top, \tag{4.7}$$

where $\boldsymbol{\Sigma}_\phi$ is the covariance matrix of $\boldsymbol{\phi}(t)$.

The objective of BSS is to recover $\mathbf{s}(t)$ from the observation $\mathbf{x}(t)$ so that its correlation satisfies $\boldsymbol{\Sigma}_{s_i,s_j} \triangleq \mathbb{E}[\bar{\boldsymbol{\Sigma}}_{s_i,s_j}] = \mathbf{0}_{n\times n}$ for $i \neq j$. However, due to the finite sample size, it does not hold for its empirical counterpart, i.e., $\bar{\boldsymbol{\Sigma}}_{s_i,s_j} \neq \mathbf{0}_{n\times n}$. In this chapter, the coefficient matrix $\mathbf{W}$ is assumed to be given, thus the linear model (4.3) implies the relation of the covariance matrices $\boldsymbol{\Sigma}_\phi = \mathbf{W}^{-}\boldsymbol{\Sigma}_s\mathbf{W}^{-\top}$ and their empirical counterpart $\bar{\boldsymbol{\Sigma}}_\phi = \mathbf{W}^{-}\bar{\boldsymbol{\Sigma}}_s\mathbf{W}^{-\top}$. The notation is denoted as $\mathbf{W}^{-} = \mathbf{W}^{-1}$ for simple expression using the following content.

In practice, only the samples of the finite size are available that lead to $\widehat{\mathrm{MSE}}$ on (4.6). To obtain the MSE of the infinite sample size, we take the mathematical expectations

$$\mathrm{MSE} = \mathbb{E}\{\widehat{\mathrm{MSE}}\} = \mathrm{tr}\left\{\boldsymbol{\Sigma}_{\delta_\phi}\mathrm{Cov}(\mathbf{W}^\top)\right\} + \mathcal{O}\left(\frac{1}{\sqrt{T}}\right). \tag{4.8}$$

The detail derivation can be found in Appendix A. The objective of this chapter is to evaluate the performance of the mismatched estimator, as well as proposing a formalization to the performance loss. This can be formulated by a contrast function.

Problem 1. *Given a set of true projected data $\boldsymbol{\phi}(t)$ that is a fixed point of the theoretical ViNLisem algorithm. If the empirical ViNLisem algorithm has an almost surely fixed point $\hat{\boldsymbol{\phi}}(t)$ that is a neighborhood of $\boldsymbol{\phi}(t)$. Then the problem is to learn an algebraic formalization that leads to the upper bound of the following equation*

$$\left\|\mathrm{MSE} - \widehat{\mathrm{MSE}}\right\|^2 = \left\|\mathrm{tr}\big\{(\boldsymbol{\Sigma}_{\delta_\phi} - \bar{\boldsymbol{\Sigma}}_{\delta_\phi})\mathrm{Cov}(\mathbf{W}^\top)\big\}\right\|^2, \tag{4.9}$$

where $\boldsymbol{\Sigma}_{\delta_\phi}$ is the covariance matrix of δ_ϕ and $\bar{\boldsymbol{\Sigma}}_{\delta_\phi}$ is the corresponding empirical counterpart. □

Problem 1 implies that if we obtain a closed-form expression on $\|\mathrm{MSE} - \widehat{\mathrm{MSE}}\|^2$, the performance loss of recovered sources can be minimized approximately by reducing the discrepancy between the $\boldsymbol{\Sigma}_{\delta_\phi}$ and its empirical counterpart. In other words, $\hat{\boldsymbol{\phi}}(t)$ is expected to extract the nonlinearity of the mixing function so as the MSE can be minimized as the sample size tends to be infinity.

The performance analysis of (4.9) can be concluded from the derivation of two parts. One is to derive an iterative expression of the coefficient matrix $\mathbf{W}$ that to be estimated are modeled as deterministic but depend on the data. The detailed derivation is described in Section 3. Another part showed in Section 4 aims to establish a closed-form expression of discrepancy between the true model and its counterpart with the finite sample size.

4.3 Estimation of Coefficient Matrix W

We now turn to present a novel expectation-maximization (EM) algorithm to estimate the coefficient matrix $\mathbf{W}$ on (4.9). In the E-step, the hidden parameter is estimated using the Kullback-Leibler divergence [129], and in M-step, the coefficient matrix $\mathbf{W}$ is updated by an online recursive approach.

4.3.1 Maximum-Likelihood (ML) Estimation

Consider a set of true projected signals $\{\boldsymbol{\phi}(\mathbf{x}(t))\}_{t=1}^T$ that are assumed to be drawn independently from a multivariate Gaussian distribution. We thus can estimate the parameters by ML estimation. The log-likelihood function is given by

$$\begin{aligned}\log p\left(\{\boldsymbol{\phi}(\mathbf{x}(t))\}_{t=1}^T\right) &= \log \prod_{t=1}^T p(\boldsymbol{\phi}(\mathbf{x}(t))) \\ &= -\frac{T}{2}\log\det(\boldsymbol{\Sigma}_\phi) - \frac{1}{2}\sum_{t=1}^T \phi(t)^\top \boldsymbol{\Sigma}_\phi^- \phi(t) - \frac{nT}{2}\log 2\pi,\end{aligned} \tag{4.10}$$

where $\det(\boldsymbol{\Sigma}_\phi)$ indicates the determinant of $\boldsymbol{\Sigma}_\phi$. The first equality comes from the assumption of independence of sources $\boldsymbol{\phi}(t)$ and $\boldsymbol{\phi}(t')$ for $t \neq t'$. The relative work has been discussed in [51]. The second equality follows the Gaussian distribution with the zero vector mean and the

covariance matrix is denoted as $\mathbf{\Sigma}_\phi$. The above equation can be rewritten by using the trace trick

$$\begin{aligned}\log p\left(\{\boldsymbol{\phi}(t)\}_{t=1}^{T}\right) &= -\frac{T}{2}\log\det(\mathbf{\Sigma}_\phi) - \frac{1}{2}\sum_{t=1}^{T}\mathrm{tr}\left\{\mathbf{\Sigma}_\phi^{-}\boldsymbol{\phi}(t)\boldsymbol{\phi}(t)^\top\right\} - \frac{nT}{2}\log 2\pi \\ &= -\frac{T}{2}\log\det(\mathbf{\Sigma}_\phi) - \frac{T}{2}\mathrm{tr}\left\{\bar{\mathbf{\Sigma}}_\phi\mathbf{\Sigma}_\phi^{-}\right\} - \frac{nT}{2}\log 2\pi \\ &= -\frac{T}{2}\log\frac{\det(\mathbf{\Sigma}_\phi)}{\det(\bar{\mathbf{\Sigma}}_\phi)} - \frac{T}{2}\mathrm{tr}\left\{\bar{\mathbf{\Sigma}}_\phi\mathbf{\Sigma}_\phi^{-}\right\} - \kappa_1, \end{aligned} \tag{4.11}$$

where $\kappa_1 = -\frac{T}{2}\log\det(\bar{\mathbf{\Sigma}}_\phi) - \frac{nT}{2}\log 2\pi$ denotes the term, which is irrelevant to the maximization of the likelihood with respect to its parameters. The results of the first equality come from the property $\mathbf{a}^\top\mathbf{\Sigma}\mathbf{a} = \mathrm{tr}\{\mathbf{\Sigma}\mathbf{a}\mathbf{a}^\top\}$ for any vector $\mathbf{a}$ and matrix $\mathbf{\Sigma}$ with appropriate dimensions. Then, using the definition of the empirical counterpart and the property of $\mathrm{tr}\{\mathbf{AB}\} = \mathrm{tr}\{\mathbf{BA}\}$, we have the second equality. To obtain a similar form with Kullback-Leibler divergence in Definition 1, the third equality is a derivation of a simple operation.

Definition 6. *Let $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ be two $n \times n$ positive definite matrices. The Kullback-Leibler divergence $\mathcal{KL}(\mathbf{\Sigma}_1\|\mathbf{\Sigma}_2)$ measures the difference between two multivariate normal distribution $\mathcal{N}(0, \mathbf{\Sigma}_1)$ and $\mathcal{N}(0, \mathbf{\Sigma}_2)$, which is given by*

$$\mathcal{KL}(\mathbf{\Sigma}_1\|\mathbf{\Sigma}_2) \stackrel{\triangle}{=} \frac{1}{2}\left(\mathrm{tr}\{\mathbf{\Sigma}_1\mathbf{\Sigma}_2^{-}\} - \log\frac{\det(\mathbf{\Sigma}_1)}{\det(\mathbf{\Sigma}_2)} - n\right). \tag{4.12}$$

We set $\mathbf{\Sigma}_1 = \bar{\mathbf{\Sigma}}_\phi$ and $\mathbf{\Sigma}_2 = \mathbf{\Sigma}_\phi$. Using the definition of Kullback-Leibler divergence [129] as a measure of two matrices, the log-likelihood of (4.11) can be rewritten as

$$\begin{aligned}\log p\left(\{\boldsymbol{\phi}(t)\}_{t=1}^{T}\right) &= -T\mathcal{KL}(\bar{\mathbf{\Sigma}}_\phi\|\mathbf{\Sigma}_\phi) - \frac{nT}{2} - \kappa_1 \\ &= -T\mathcal{KL}(\mathbf{W}\bar{\mathbf{\Sigma}}_s\mathbf{W}^\top\|\mathbf{\Sigma}_\phi) + \kappa_2, \end{aligned} \tag{4.13}$$

where $\kappa_2 = -\frac{T}{2}\log\det(\bar{\mathbf{\Sigma}}_\phi) - \frac{nT}{2}\log 2\pi - \frac{nT}{2}$ denotes the term which is irrelevant to the maximization of the likelihood with respect to its parameters. The stationary point of Kullback-Leibler divergence in (4.13) leads to the way for estimating the covariance matrix $\mathbf{\Sigma}_\phi$.

4.3.2 Estimation of Σ_ϕ for a Fixed $\mathbf{W}$

To analyze $\mathbf{\Sigma}_\phi$, we fix the coefficient matrix $\mathbf{W}$ first. Thus, maximizing log-likelihood (4.11) is equivalent to minimizing the Kullback-Leibler divergence, which is given by

$$\max_{\mathbf{\Sigma}_\phi}\log p\left(\{\boldsymbol{\phi}(t)\}_{t=1}^{T}|\mathbf{\Sigma}_\phi, \mathbf{W}\right) = \min_{\mathbf{\Sigma}_\phi}\mathcal{KL}(\mathbf{W}\bar{\mathbf{\Sigma}}_s\mathbf{W}^\top\|\mathbf{\Sigma}_\phi). \tag{4.14}$$

To derive an estimator $\mathbf{\Sigma}_\phi$ for the fixed $\mathbf{W}$, the definition of Kullback-Leibler divergence in (4.12) can be rewritten in the form as

$$\begin{aligned}\mathcal{KL}(\mathbf{\Sigma}_1\|\mathbf{\Sigma}_2) &= \frac{1}{2}\mathrm{tr}\{\mathbf{\Sigma}_1\mathbf{\Sigma}_2^{-} - \mathbf{I}\} - \frac{1}{2}\log\det(\mathbf{\Sigma}_1\mathbf{\Sigma}_2^{-}) \\ &= \frac{1}{2}\mathrm{tr}\{\mathbf{\Sigma}_2^{-\frac{1}{2}}\mathbf{\Sigma}_1\mathbf{\Sigma}_2^{-\frac{1}{2}} - \mathbf{I}\} - \frac{1}{2}\log\det(\mathbf{\Sigma}_2^{-\frac{1}{2}}\mathbf{\Sigma}_1\mathbf{\Sigma}_2^{-\frac{1}{2}}) \end{aligned}$$

$$= \frac{1}{2}\mathrm{tr}\{\boldsymbol{\mathcal{R}} - \mathbf{I}\} - \frac{1}{2}\log\det(\boldsymbol{\mathcal{R}}). \tag{4.15}$$

We note that the fact, for a positive-definite matrix $\boldsymbol{\mathcal{R}}$, $\mathrm{tr}(\boldsymbol{\mathcal{R}} - \mathbf{I})$ is an upper bound for $\log\det\boldsymbol{\mathcal{R}}$, which is attained if and only if $\boldsymbol{\mathcal{R}}$ is the identity. This follows immediately from the inequality $\log x \leq x - 1$ which is valid for all $x > 0$. Thus, the equality has the minimization value if and only if $x = 1$.

Therefore, minimizing the right-hand side of (4.14) can be understood as diagonalization of covariance matrix $\bar{\boldsymbol{\Sigma}}_\phi$ by matrix $\mathbf{W}$. Since we assume the source signals $\boldsymbol{\phi}(t)$ are independent of $\boldsymbol{\phi}(t')$ for any $t \neq t'$, then we have the covariance matrix $\boldsymbol{\Sigma}_{\phi_i,\phi_j} = \mathbf{0}_{n\times n}$ for $i \neq j$. That is, $\mathcal{KL}(\mathbf{W}\bar{\boldsymbol{\Sigma}}_s\mathbf{W}^\top\|\boldsymbol{\Sigma}_\phi) \geq 0$ is satisfied with equality if and only if $\boldsymbol{\Sigma}_\phi^{ML} = \mathrm{diag}\{\mathbf{W}\bar{\boldsymbol{\Sigma}}_s\mathbf{W}^\top\}$, where $\mathrm{diag}\{\cdot\}$ is the operator of the diagonalization. Then (4.14) takes the form

$$\max_{\boldsymbol{\Sigma}_\phi} \log p(\{\boldsymbol{\phi}(t)\}_{t=1}^T|\boldsymbol{\Sigma}_\phi, \mathbf{W}) = -T\mathcal{KL}(\mathbf{W}\bar{\boldsymbol{\Sigma}}_s\mathbf{W}^\top\|\mathrm{diag}\{\mathbf{W}\bar{\boldsymbol{\Sigma}}_s\mathbf{W}^\top\} + \kappa_2, \tag{4.16}$$

where $\kappa_2 = -\frac{T}{2}\log\det(\bar{\boldsymbol{\Sigma}}_\phi) - \frac{nT}{2}\log 2\pi - \frac{nT}{2}$ is a irrelevant term with respect to the parameter. $\bar{\boldsymbol{\Sigma}}_\phi$ is an empirical counterpart that can be determined by the data.

4.3.3 Estimate W for a Fixed $\boldsymbol{\Sigma}_\phi$

Now let us consider to estimate the coefficient matrix $\mathbf{W}$ by characterizing an iterative expression, where the hidden parameter $\boldsymbol{\Sigma}_\phi$ is used to update the coefficient matrix $\mathbf{W}$. For this purpose, we calculate the derivative of

$$\begin{aligned}\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_\phi) &= \mathcal{KL}(\mathbf{W}\bar{\boldsymbol{\Sigma}}_s\mathbf{W}^\top\|\boldsymbol{\Sigma}_\phi), \\ &= \frac{1}{2}\left(\mathrm{tr}\{\mathbf{W}\bar{\boldsymbol{\Sigma}}_s\mathbf{W}^\top\boldsymbol{\Sigma}_\phi^-\} - \log\frac{\det(\mathbf{W}\bar{\boldsymbol{\Sigma}}_s\mathbf{W}^\top)}{\det(\boldsymbol{\Sigma}_\phi)} - n\right),\end{aligned} \tag{4.17}$$

with respect to $\mathbf{W}$ for fixed $\boldsymbol{\Sigma}_\phi$. The derivative of Kullback-Leibler divergence can be computed using the first-order variation of $\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_\phi)$ when $\mathbf{W}$ is replaced by $\mathbf{W}(\mathbf{I} + \mathbf{D})$ in the Taylor expansion

$$\mathcal{J}(\mathbf{W}(\mathbf{I}+\mathbf{D}), \boldsymbol{\Sigma}_\phi) = \mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_\phi) + \mathrm{tr}\left\{(\nabla\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_\phi))^\top\mathbf{DW}\right\} + \mathcal{O}(\mathbf{DW}), \tag{4.18}$$

where the multiplier $\mathbf{D}$ or matrix $\nabla\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_\phi))$ is called the relative gradient [14, 5] of $\mathcal{J}(\mathbf{W}(\mathbf{I}+\mathbf{D}), \boldsymbol{\Sigma}_\phi)$ with respect to $\mathbf{W}$.

$$\mathcal{J}(\mathbf{W} + \mathbf{DW}) = \mathcal{J}(\mathbf{W}) + \mathrm{tr}\left\{(\frac{\partial\mathcal{J}}{\partial\mathbf{W}})^\top\mathbf{DW}\right\} + \text{higher-order in } \mathbf{D}. \tag{4.19}$$

The largest decrement in the value of $\mathcal{J}(\mathbf{W}(\mathbf{I} + \mathbf{D}) - \mathcal{J}(\mathbf{W})$ is now obviously obtained when the term $\mathrm{tr}\{\nabla\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_\phi)\mathbf{W}^\top)^\top\mathbf{D}\}$ is minimized, which happens when $\mathbf{D}$ is proportional to $-\nabla\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_\phi)\mathbf{W}^\top$. The coefficient matrix $\mathbf{W}$ is updated sequentially by using the steepest descent method

$$\mathbf{W} \leftarrow \mathbf{W} - \nabla\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_\phi)\mathbf{W}^\top\mathbf{W}, \tag{4.20}$$

Algorithm 2 Estimate coefficient matrix $\mathbf{W}$ using an iterative algorithm

Initialization: Choose initial estimations $\mathbf{W}^{(0)}$ and the value of the threshold ϵ.

for $t = 1$ **do**
  E-step: Compute $\mathbf{\Sigma}_{\phi}^{(t)}$ from
  $\mathbf{\Sigma}_{\phi}^{(t)} = \operatorname{diag}\{\mathbf{W}^{(t-1)}\bar{\mathbf{\Sigma}}_s(\mathbf{W}^{(t-1)})^{\top}\}$
  M-step: Update $\mathbf{W}^{(t)}$ using
  $\mathbf{W}^{*(t)} = \left[2\mathbf{I} - \frac{1}{2}(\mathbf{\Sigma}_{\phi}^{(t)})^{-}(\bar{\mathbf{\Sigma}}_{\phi}^{\top} + \bar{\mathbf{\Sigma}}_{\phi})\right]\mathbf{W}^{(t-1)}$
  $\mathbf{W}^{(t)} = \frac{\mathbf{W}^{*(t)}}{\|\mathbf{W}^{*(t)}\|}$
  Check for convergence
  if $\|\mathbf{W}^{(t)} - \mathbf{W}^{(t-1)}\| \leq \epsilon$ **then**
    Break
  else
    $t = t + 1$
  end if
end for

where the symbol $\leftarrow$ means substitution, i.e., the value of the right-hand side is computed and substituted on the left-hand side. The natural gradient method [10, 130] is suggested for updating with a faster convergence, which is given by

$$\mathbf{W} \leftarrow \mathbf{W} + \left[\mathbf{I} - \frac{1}{2}\mathbf{\Sigma}_{\phi}^{-}(\bar{\mathbf{\Sigma}}_{\phi}^{\top} + \bar{\mathbf{\Sigma}}_{\phi})\right]\mathbf{W}. \tag{4.21}$$

The detailed derivation is shown in Appendix B. Therefore, the approach provides a convergence point of the maximum likelihood estimation by a recursive approach.

4.4 The Estimation for Covariance Matrix

In this section, we will introduce a procedure of establishing a closed-form expression for bounding the covariance matrix $\bar{\mathbf{\Sigma}}_{\delta_\phi}$ under both the operator norm and a class of tapering estimator [128].

Theorem 4. *Let $\bar{\mathbf{\Sigma}}_{\delta_\phi}$ be an estimator of the $k \times k$ covariance matrix $\mathbf{\Sigma}_{\delta_\phi}$ on the finite sample. For $k \geq T^{1/(2\alpha+1)}$*

$$\mathbb{E}\left\|\bar{\mathbf{\Sigma}}_{\delta_\phi} - \mathbf{\Sigma}_{\delta_\phi}\right\|^2 \leq C_2\frac{m + \log k}{T} + C_1^2\left(\frac{p}{2}\right)^{-2\alpha}. \tag{4.22}$$

In above equation, k is the dimension of the data in the feature space that the parameter k is required to satisfy $k \leq T^{\frac{1}{2\alpha+1}}$, where T is the number of the samples and parameter α is selected to establish the above inequality. In our experiments, the parameter α and p are selected according to $k \leq T^{\frac{1}{2\alpha+1}}$ and $1 \leq p \leq k$. The parameter m is an integer according to $0 < m \leq k$. □

The above problem can be analyzed by dividing into the squared bias and the squared variance as

$$\mathbb{E}\|\bar{\mathbf{\Sigma}}_{\delta_\phi} - \mathbf{\Sigma}_{\delta_\phi}\|^2 \leq \underbrace{\|\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi}\|^2}_{bias} + \underbrace{\mathbb{E}\|\bar{\mathbf{\Sigma}}_{\delta_\phi} - \mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}]\|^2}_{variance}. \tag{4.23}$$

The detailed derivation is shown in Appendix C. The first term is called as the squared bias, which represents the extent to that the average estimation over all data sets differs from the desired approximation function [131]. The second term is called the variance, which measures the extent to that the approximated model for individual data sets varies around their average. Hence this measures the extent to which the function $\bar{\mathbf{\Sigma}}_{\delta_\phi}$ is sensitive to the particular choice of data set.

In the remainder of this section, we shall provide asymptotic analysis for the bias and the variance, and use these to investigate how the (4.9) will behave.

4.4.1 Bias Analysis

We begin by presenting the risk upper bound for the squared bias. The derivation of the procedure is inspired by the idea of convergence bound under the spectral norm [132].

Definition 7. *Let $\rho(\mathbf{A})$ be a spectral radius of $\mathbf{A}$. If $\mathbf{A} \in \mathbb{R}^{k\times k}$ is a symmetric matrix with eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_k$, then $\|\mathbf{A}\|_2 = \rho(\mathbf{A})$ has the definition as*

$$\|\mathbf{A}\|_2 = \rho(\mathbf{A}) \stackrel{\triangle}{=} \max_{1\leq i\leq k}\{|\lambda_i|\}. \tag{4.24}$$

□

Definition 8. *For any matrix $\mathbf{A} \in \mathbb{R}^{k\times k}$ of size $k \times k$, the matrix norm $\|\mathbf{A}\|$ is defined by $\|\mathbf{A}\| := \max_{1\leq i\leq k} \sum_{j=1}^{k} |a_{ij}|$.* □

Theorem 5. *For a symmetric matrix $\mathbf{A} \in \mathbb{R}^{k\times k}$, $\|\mathbf{A}\|_2$ is bounded by matrix norm in the terms of*

$$\|\mathbf{A}\|_2 \leq \|\mathbf{A}\| \stackrel{\triangle}{=} \max_{1\leq i\leq k} \sum_{j=1}^{k} |a_{ij}|. \tag{4.25}$$

□

Proof. Assume λ_i is an arbitrary eigenvalue of matrix $\mathbf{A}$ and $\mathbf{v}_i$ is its corresponding eigenvector. Then we have

$$|\lambda_i|\|\mathbf{v}_i\| = \|\lambda_i \mathbf{v}_i\| = \|\mathbf{A}\mathbf{v}_i\| \leq \|\mathbf{A}\|\|\mathbf{v}_i\|. \tag{4.26}$$

Since $\mathbf{v}_i$ is a non-zero vector $\|\mathbf{v}_i\| \neq 0$, then $|\lambda_i| \leq \|\mathbf{A}\|$. Since λ_i is an arbitrary eigenvalue, then we have $\rho(\mathbf{A}) \leq \|\mathbf{A}\|_1$. □

This result is considered to be used for the convergence of the bias part in (4.23) as $\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi}$ by its l_1 norm.

Considering infinite sample size, the convariance matrix $\mathbf{\Sigma}_{\delta_\phi}$ is defined by

$$\mathbf{\Sigma}_{\delta_\phi} = (\sigma_{ij})_{1\leq i,j\leq k} = \mathbb{E}\left[\boldsymbol{\delta}_\phi(t)\boldsymbol{\delta}_\phi(t)^\top\right], \tag{4.27}$$

where σ_{ij} is the element of the convariance matrix. Since the data points are finite in practice, the definition of empirical counterpart using the estimator is given by

$$\bar{\mathbf{\Sigma}}_{\delta_\phi} = (\omega_{ij}\sigma_{ij})_{k\times k} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\delta}_\phi(t)\boldsymbol{\delta}_{\boldsymbol{\phi}}(t)^\top, \tag{4.28}$$

where ω_{ij} is the weight. Without loss of generality, the weight ω_{ij} can be defined as

$$\omega_{ij} = \begin{cases} 1, & \text{when } |i-j| < \frac{p}{2}, \\ 2 - \frac{2|i-j|}{p}, & \text{when } \frac{p}{2} \leq |i-j| < p, \\ 0, & \text{otherwise.} \end{cases} \tag{4.29}$$

where p is an even integer with $1 \leq p \leq k$. As noted in [128], the estimated covariance matrices $\mathbf{\Sigma}_{k\times k} = [\sigma_{ij}]_{1\leq i,j\leq k}$ can be considered over the following parameter space.

$$\left\{\mathbf{\Sigma} : \max_i \sum_j \left\{|\sigma_{ij}| : |i-j| \geq \frac{p}{2}\right\} \leq C_1\left(\frac{p}{2}\right)^{-\alpha}\right\}. \tag{4.30}$$

The parameter α essentially specifies the rate of decay for the covariances σ_{ij} as they move away from the diagonal, where α can be viewed as an analog of the smoothness parameter in non-parametric function estimation problems.

Thus, the bias part $\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi}$ can be expressed in the form of

$$\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi} = \left[(\omega_{ij}-1)\sigma_{ij}\right]_{k\times k}, \tag{4.31}$$

where $\omega_{ij} \in [0,1]$. Since the operator norm of a symmetric matrix is bounded by its l_1 norm, in which $\omega_{ij} = 1$ when $|i-j| < \frac{p}{2}$, then

$$\left\|\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi}\right\|^2 = \max_{1\leq i\leq k}\left[\sum_{j:\frac{p}{2}\leq|i-j|<p}\left|\left(1-\frac{2|i-j|}{p}\right)\sigma_{ij}\right| + \sum_{j:p\leq|i-j|}|\sigma_{ij}|\right]^2. \tag{4.32}$$

The detailed derivation can be found in Appendix D. If $\frac{p}{2} \leq |i-j| < p$, we have $-1 < 1-\frac{2|i-j|}{p} \leq 0$. Thus, we have $\sum_{j:\frac{p}{2}\leq|i-j|<p}\left|(1-\frac{2|i-j|}{p})\sigma_{ij}\right| \leq \sum_{j:\frac{p}{2}\leq|i-j|<p}|\sigma_{ij}|$. Then the above equation can be written as

$$\begin{aligned}\left\|\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi}\right\|^2 &\leq \left[\max_{1\leq i\leq k}\sum_{j:|i-j|\geq\frac{p}{2}}|\sigma_{ij}|\right]^2 \\ &\leq C_1^2\left(\frac{p}{2}\right)^{-2\alpha}.\end{aligned} \tag{4.33}$$

Here, $\alpha > 0$ indicates the optimal rate of convergence for the estimator $\bar{\mathbf{\Sigma}}_{\delta_\phi}$.

4.4.2 Variance Analysis

Next, we consider the upper bound of the squared variance. The derivation of the idea is inspired by a special class of tapering estimator [128]. The approach employs a matrix as the sum of many small block matrices along the diagonal, where the block matrices are given by

$$\mathbf{M}_l^{(m)} = (\sigma_{ij}\Omega\{l \le i < l+m, l \le j < l+m\})_{k\times k}\,, \tag{4.34}$$

where $\Omega\{l \le i < l+m, l \le j < l+m\}$ is an indicator that assigns the value one to the elements in this range of matrix. Without loss of generality, we assume that k is divisible by m. By setting $\mathbf{S}^{(m)}$ as $\mathbf{S}^{(m)} = \sum_{l=1-m}^{k} \mathbf{M}_l^{(m)}$, the tapering estimator can be written as

$$\hat{\mathbf{\Sigma}}_{\delta_\phi}^{(m)} = \frac{1}{m_h}(\mathbf{S}^{(m)} - \mathbf{S}^{(m_h)}), \tag{4.35}$$

where $m_h = \frac{m}{2}$. The performance of the estimator $\hat{\mathbf{\Sigma}}_{\delta_\phi}^{(m)}$ depends on the choice of the parameter m. From the above equation, we can see that the estimator $\hat{\mathbf{\Sigma}}_{\delta_\phi}^{(m)}$ can be written as the sum of a large number of small disjoint block matrices.

Lemma 1. *Let $\hat{\mathbf{\Sigma}}_{\delta_\phi}^{(m)}$ be an estimator, which is defined in (4.35). Then we have*

$$\left\| \hat{\mathbf{\Sigma}}_{\delta_\phi}^{(m)} - \mathbb{E}[\hat{\mathbf{\Sigma}}_{\delta_\phi}^{(m)}] \right\| \le 3\mathcal{N}_l^{(m)}, \tag{4.36}$$

where $\mathcal{N}_l^{(m)} = \max_{1\le l\le k} \|\mathbf{M}_l^{(m)} - \mathbb{E}[\mathbf{M}_l^{(m)}]\|$. □

The proof derivation can be found in Appendix E.

Lemma 2. *Assume that the distribution of $\phi(x_i)$ is sub-Gaussian in the sense that there is $\rho > 0$ such that*

$$\mathbb{P}\{|\mathbf{v}^\top(\boldsymbol{\phi}(\mathbf{x}_i) - \mathbb{E}[\boldsymbol{\phi}(\mathbf{x}_i)])| > t\} \le \exp(-t^2\rho/2), \tag{4.37}$$

where $t > 0$ and $\|\mathbf{v}\|_2 = 1$. Then there is a constant $\rho_1 > 0$ such that

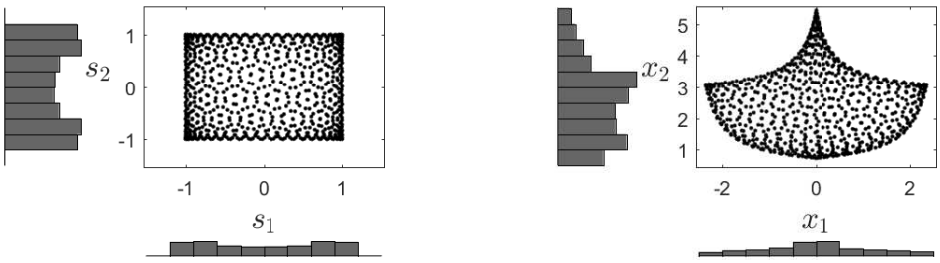
$$\mathbb{P}\{\mathcal{N}_l^{(m)} > 0\} \le 2k5^m \exp(-nx^2\rho_1), \tag{4.38}$$

for all $0 < x < \rho_1$ and $1 - m \le l \le p$. □

The detailed derivation can be found in Appendix F. Since x is bounded as $0 < x < \rho_1$, then we have $\mathbb{E}\|\hat{\mathbf{\Sigma}}_{\delta_\phi}^{(m)} - \mathbb{E}[\hat{\mathbf{\Sigma}}_{\delta_\phi}^{(m)}]\|^2$ is bounded by a constant.

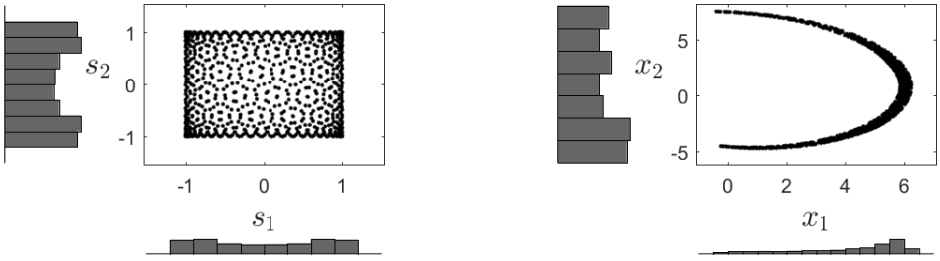
4.5 Simulation Results

In this section, we present some illustrative examples to demonstrate the validity of the computed bounds.



(a) Source signals (b) Mixture signals

Figure 4.2: (a) The source signals are generated from the artificial data of two sinusoidal signals. (b) The mixture signals are nonlinearly mixed by the DS mixture function.



(a) Source signals (b) Mixture signals

Figure 4.3: (a) The source signals are generated from the artificial data of two sinusoidal signals. (b) The mixture signals are nonlinearly mixed by the PNL function.

Without loss of generality, the observed signals can be standardized [133] by the data centering and whitening in terms of

$$\tilde{\mathbf{x}} := \mathrm{Cov}(\mathbf{x})^{-\frac{1}{2}}(\mathbf{x} - \mathbb{E}[\mathbf{x}]), \tag{4.39}$$

where the standardized signal clearly satisfies $\mathbb{E}[\tilde{\mathbf{x}}] = 0$ and $\mathrm{Cov}(\tilde{\mathbf{x}}) = \mathbf{I}$. For finite sample size, the standardization procedure (4.39) can be carried out empirically. The estimators of $\mathbb{E}[\mathbf{x}]$ and $\mathrm{Cov}(\mathbf{x})$ take respectively the mean and covariance of the matrix $\mathbf{x}$ empirically as

$$\bar{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}(t), \quad \bar{\mathbf{\Sigma}}_x = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}(t) - \bar{\mathbf{x}})(\mathbf{x}(t) - \bar{\mathbf{x}})^\top. \tag{4.40}$$

The empirically standardized data can then be defined as

$$\tilde{\mathbf{x}} := \bar{\mathbf{\Sigma}}_x^{-\frac{1}{2}}(\mathbf{x}(t) - \bar{\mathbf{x}}). \tag{4.41}$$

Note that the whitening procedure is used as a preprocessing for all the following experiments in the convention.

The performances are studied by using estimated and natural data in terms of the mean squared error (MSE) in [52], which is given by

$$\mathrm{MSE}(\mathbf{s}_i, \hat{\mathbf{s}}_i) = 10\log_{10}\left(\frac{1}{n}\sum_{i=1}^{n}\min_{\delta}\|\mathbf{s}_i - \delta\hat{\mathbf{s}}_i\|_2^2\right), \tag{4.42}$$

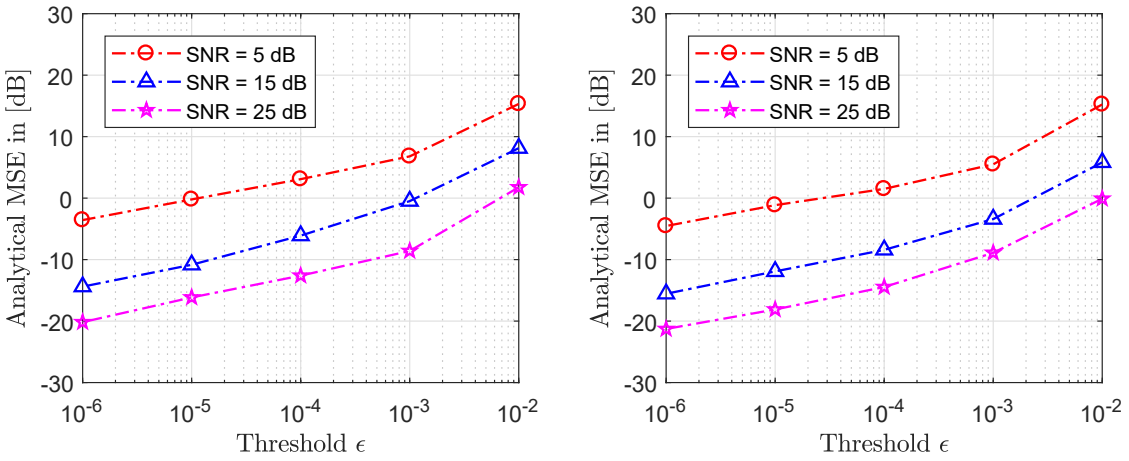


Figure 4.4: The Analytical MSE versus the Different Values of Threshold ϵ. The top figure uses the observations that are nonlinearly mixed by the DS function of (4.43). The bottom figure is the observations that are nonlinearly mixed by the PNL function of (4.45).

Table 4.1: Descriptions of Real-World Data [1].

Name	Scenario	Duration(s)	Sample Size	Overlap
AMI[1]	News	6	50,000	yes
CHiME3[2]	Talker	6	50,000	yes
Nonspeech[3]	TV order	10	160,000	no
Multitrack[4]	Theater	147	6,482,701	yes

where $\hat{\mathbf{s}}_i$ denotes the estimate of the source signal $\mathbf{s}_i$, and δ is a scalar reflecting the scalar ambiguity.

In addition, parameter determination is still an open problem [134]. The closed-form expressions of MSE in (4.22) depends on the choice of parameter. We determine the parameter $\alpha = 0.1$ and other parameters, such as $p = \lfloor T^{\frac{1}{2\alpha+1}} \rfloor$ are empirically determined as in traditional approaches [128].

4.5.1 Deterministic Artificial Data

In the first group, we generate the data points from two sinusoidal signals that have different frequencies, such that $s_1(t) = \sin(0.05\pi t)$ and $s_2(t) = \sin(0.021\pi t)$ that was used in [42, 43, 44]. These source signals are nonlinearly mixed by two kinds of nonlinear mixture functions, including the distorted source (DS) in [104], and the post-nonlinear mixture (PNL) in [31, 33].

In the distorted source (DS) mixture function of (4.43), each observation is a linear mixture of nonlinear distorted sources. Specifically, in the experiments the two channel mixtures were generated according to

$$\begin{aligned} x_1(t) &= \exp(s_1(t)) - \exp(s_2(t)), \\ x_2(t) &= \exp(-s_1(t)) + \exp(-s_2(t)). \end{aligned} \tag{4.43}$$

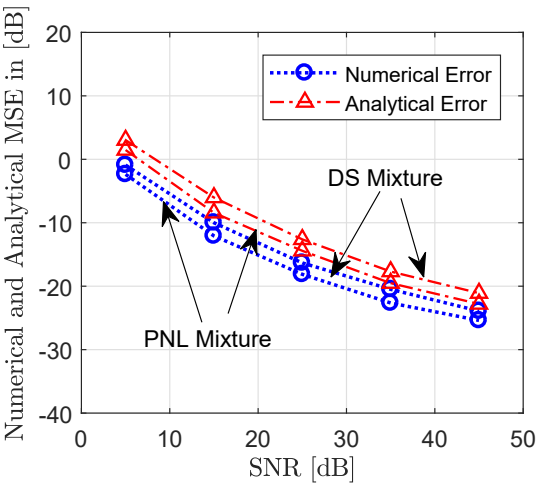


Figure 4.5: The Performance of Numerical MSE and Analytical MSE versus the Different SNR Intensities. The dash-dotted curve represents an analytical error. The dashed curve represents a numerical error.

Fig. 4.2 shows the scatter plots of the source signals $\mathbf{s}(t)$ and the mixture signals $\mathbf{x}(t)$. To see the level of nonlinear distortion after the mixing transformation, we give the scatter plot of the affinity transformation of $\mathbf{s}(t)$ in Fig. 4.2: (b).

The post-nonlinear (PNL) mixtures constitute a particularly interesting example of the theoretical separability characterized by weak indeterminacy. The sources are the first subject to a linear mixture $\mathbf{z}(t) = \mathbf{A}\mathbf{s}(t)$, where $\mathbf{A}$ is a 2×2 mixing matrix given by

$$\mathbf{A} = \begin{pmatrix} -0.2261 & -0.1189 \\ -0.1706 & -0.2836 \end{pmatrix}. \tag{4.44}$$

Then each mixture component is generated from a nonlinear, invertible transformation, as the form of

$$\begin{aligned} x_1(t) &= (z_2(t) + 3z_1(t) + 6)\cos(1.5\pi)z_1(t), \\ x_2(t) &= (z_2(t) + 3z_1(t) + 6)\sin(1.5\pi)z_1(t). \end{aligned} \tag{4.45}$$

The source signals are plotted in Fig. 4.3: (a). The mixture signals are shown in Fig. 4.3: (b), where we can see the distortions caused by the nonlinear function.

Example 1: In this example, the numerical experiments are expected to show the behavior of analytical MSE on the varying values of the threshold ϵ. The details are described in Algorithm 1. Two kinds of mixture signals are generated from (4.43) and (4.45), respectively. We fixed the sample size as $T = 1,000$. To reduce the randomness effect, 100 times Monte Carlo simulations are performed.

The analytical MSE with different setting of the threshold is exhibited in Fig. 4, where the observations are nonlinearly mixed by DS function and PNL function, respectively in the top and bottom figures. The curves are plotted for the different signal-to-noise power ratio (SNR) in decibels. As illustrated in the figure, the analytical MSE increases as the values of threshold

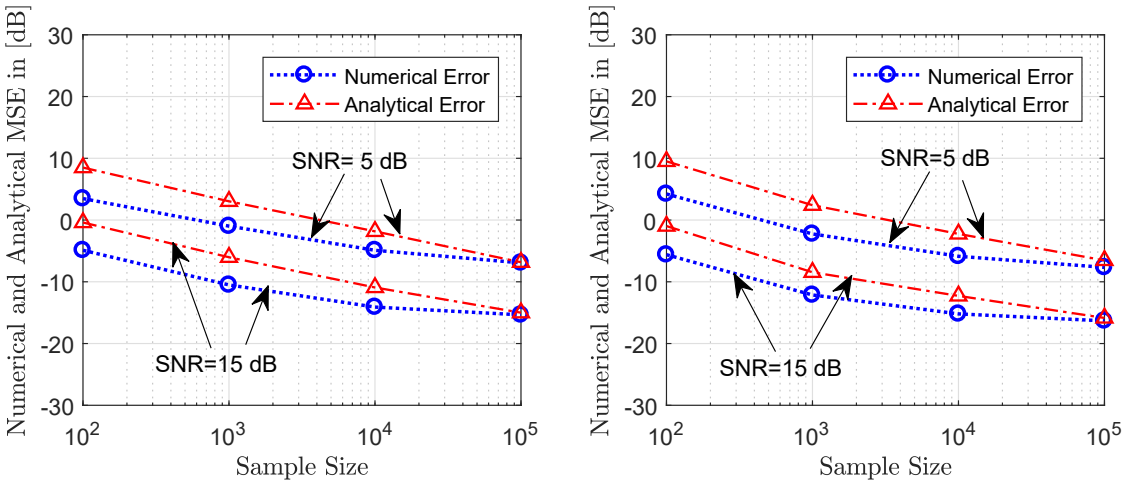


Figure 4.6: The Convergence Behavior versus the Sample Size. The dash-dotted lines represent analytical error on SNR= 5 dB. The dashed lines represent numerical error on SNR= 15 dB. The data used for the top figure are generated from DS mixture function. The data used for the bottom figure are generated from PNL mixture function.

increases. Thus, the difference between two consecutive coefficient matrices $\mathbf{W}$ converges to the threshold. Besides, the asymptotic conditions are reached even for a small threshold.

Example 2: This example contains the comparison of numerical and analytical MSEs on two kinds of artificial data with different noise intensities. The threshold is set as $\epsilon = 10^{-4}$. Both numerical and analytical MSEs are evaluated for the different SNR that varies from 5 dB to 45 dB with the step of 10 dB. We repeated the trials for 20 times and plotted the average results in Fig. 5, where the numerical and analytical errors are marked as dash-dotted line and dashed line, respectively. As seen from the figure, the analytical errors approach to the numerical errors for both the DS mixture and the PNL mixture with different settings of SNR. The numerical error decreases as the values of SNR increases, with the largest rate of the decrease occurring when the SNR in 45 dB, which lead to the lowest value of the numerical land analytical curves.

Example 3: This example compares the numerical error and analytical error with different setting of sample size. The threshold is set as $\epsilon = 10^{-4}$. The simulation is on the different value of SNR, such as 5 dB and 15 dB. To reduce the randomness effect, 20 times of Monte Carlo simulations are performed. As can be seen from Fig. 6, both the numerical error and the analytical error tend to be smaller as the number of samples increases. Moreover, the numerical error approaches to the analytical error with the number of samples increasing.

4.5.2 Real-World Audio Data

To evaluate the analytical MSE, the experiments are repeated on several real-world datasets, which are publicly available [1]. Each dataset has its own advantages, depending on whether one is interested in a variety of environments, in the duration time, or in the sample size. For instance, the data "AMI" has two kinds of sound from the cable news and network news. Another data "Multitrack" was mixed with voices of two anonymous singers. The size of the samples was varied to assess how the amount of data affects the performance of the algorithm. The general properties of the datasets are summarized in Table 4.1.

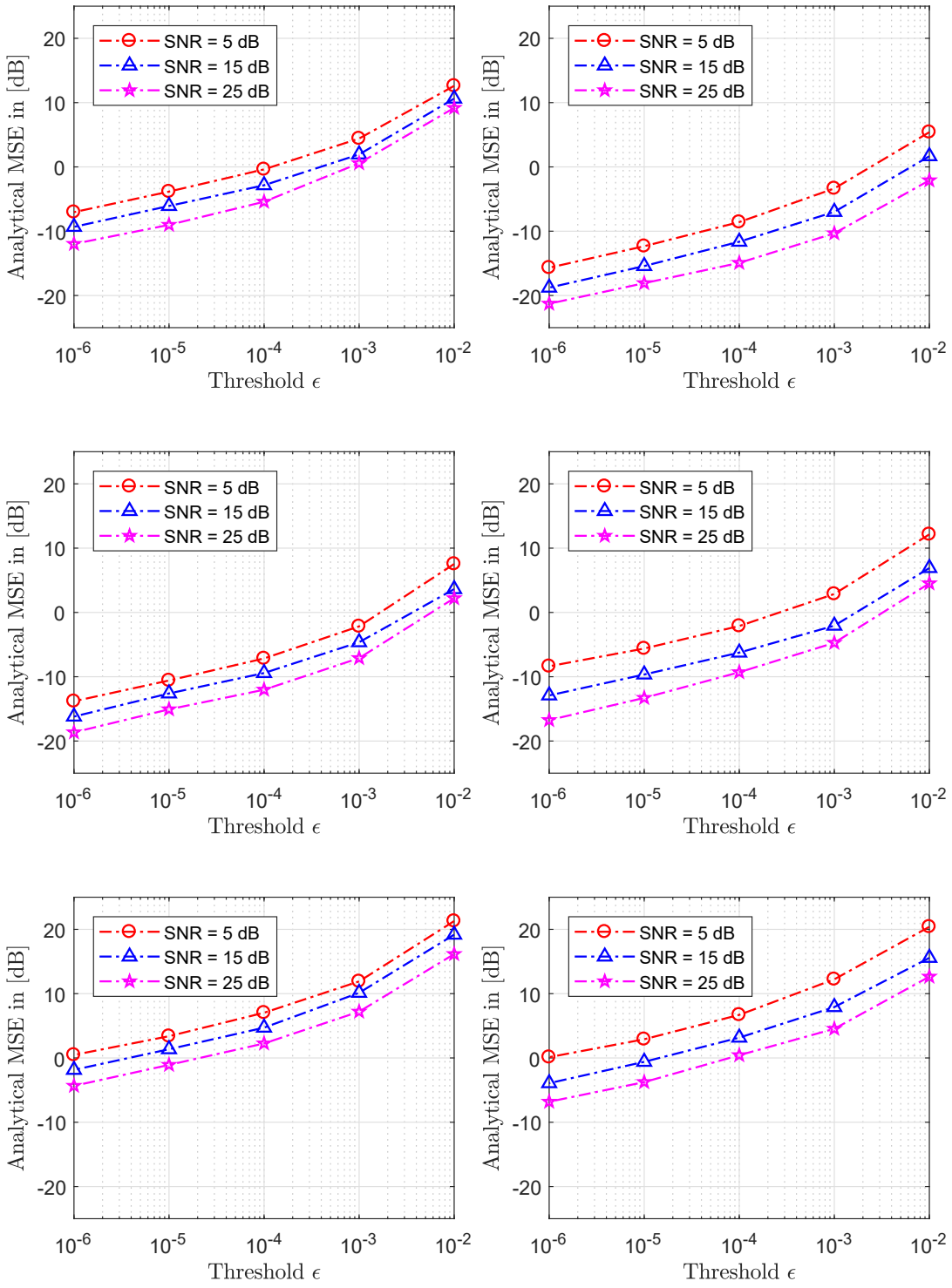


Figure 4.7: The Analytical MSE versus the Different Values of the Threshold ϵ. The results consider the real-world datasets, such as "AMI", "CHiME3" and "Nonspeech". The top figures use the observations that are nonlinearly mixed by DS function on (4.43) and PNL function on (4.45), respectively using "AMI" dataset. The middle figures use the observations that are nonlinearly mixed using the same functions on "CHiME3" dataset. The bottom figures are the observations of both DS and PNL mixture on the "Nonspeech" dataset.

In Fig. 4.7, we have shown the analytical MSE averaged over 100 times that plotted as a function of SNR. For the observations, three real-world datasets, AMI, CHiME3, and Nonspeech

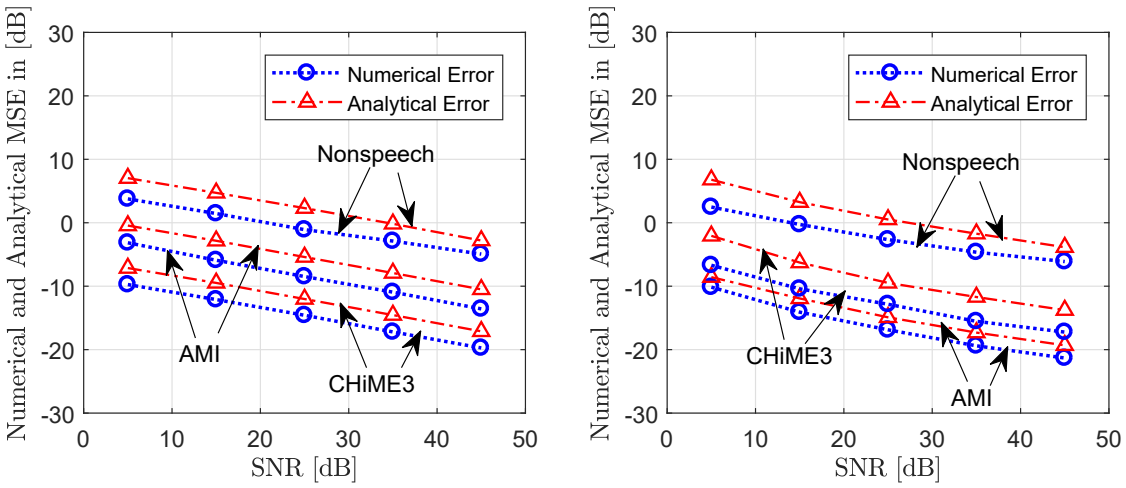


Figure 4.8: The Numerical MSE and Analytical Bound Averaged for 20 Trials versus the Different SNR Intensities. The dash-dotted lines represent analytical bound. The dashed lines represent numerical MSE. The threshold ϵ is set as 10^{-4}. The results consider the real-world datasets, such as "AMI", "CHiME3" and "Nonspeech". The left figure uses the observations that are nonlinearly mixed of three datasets respectively, by the DS function on (4.43). The right figure is the observations that are nonlinearly mixed of the same datasets by the PNL function of (4.45).

are mixed nonlinearly with the DS function and the PNL function. The size of samples for the evaluation is set as 1000. One can see that the analytical MSE has large error values for the loose threshold, however they become small as the threshold value ϵ decreases. Analytically, we need enough threshold value to have a convergence with a high level of accuracy.

Fig. 4.8 illustrates the behaviors of the numerical and analytical MSEs with the different noise intensities on the varying real-world datasets. The curves are labeled with these datasets, such as "AMI", "CHiME3", and "Nonspeech". We set the threshold as $\epsilon = 10^{-4}$ that implies a moderate stopping criterion. The performance of the three datasets shows the trend to that the curves have the lower value when the noise intensity increases. However, since the sample size is kept constant as $1,000$, the convergence with a high level of accuracy did not occur even when the SNR is 45 dB.

Fig. 4.9 exhibits the comparison of the numerical and analytical MSEs with different sample size. The datasets Nonspeech and Multitrack with big size are considered in this example. We also use a moderate threshold with the value $\epsilon = 10^{-4}$. The left column of Fig. 9 shows the comparison of results with the observations nonlinear mixed by the DS mixture function, while the right column uses the PNL mixture function. The results reflect the fact that the numerical error converges to a constant as the sample size increases.

4.6 Conclusion

In this chapter, we provide an analytical mean squared error (MSE) for the separation approach, which includes the closed-form expression of MSE as well as proposing a new algebraic formalization that leads to an upper bound on the numerical MSE. The analysis stems from the performance of a mismatched estimator that accesses the finite sample size. The idea is inspired by the derivation of two parts. One is to derive an iterative expression from the perspective of the

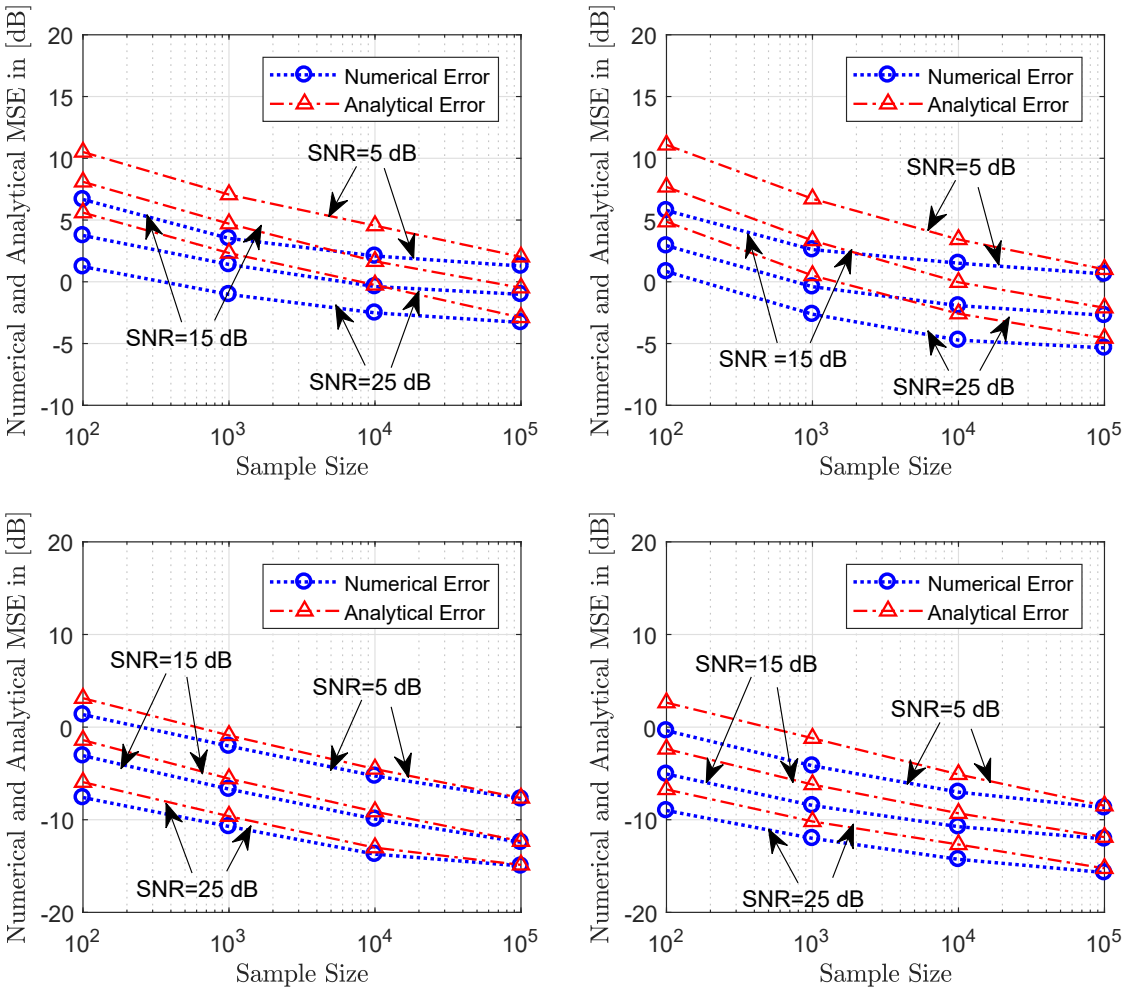


Figure 4.9: The Numerical MSE and Analytical Bound Averaged for 20 Trials versus the Different Sample Sizes. The dash-dotted lines represent analytical bound. The dashed lines represent numerical MSE. The results consider two real-world datasets "Nonspeech" and "Multitrack" with big size. The figures in the left column use the observations that the "Nonspeech" and "Multitrack" are nonlinearly mixed by the DS mixture function, respectively. Similarly, the data used for the right column are generated from the PNL mixture function.

expectation-maximization (EM) algorithm. Another one is to establish a closed-form expression for bounding the covariance matrix under both the operator norm and a special class of tapering estimators.

First, we propose a novel EM algorithm to estimate the coefficient matrix, which is modeled as deterministic but depend on the dataset. To estimate the hidden variable, the E-step used to obtain a convergence point of the maximum likelihood estimator, which could be interpreted as the stationary point that minimizes the Kullback-Leibler (KL) divergence. In the M-step, the hidden parameter is used to update the coefficient matrix by an online recursive version. Then, we establish a closed-form expression for bounding the covariance matrix, as well as measuring the non-parametric function mis-specification problems with the finite sample size. The simulation results illustrate that the trends of the numerical result follows the analytical MSE in different scenarios.

4.7 Appendix

4.7.1 Asymptotic Expression for the MSE

Without loss of generality, (4.6) can be equally rewritten as

$$\begin{aligned}\widehat{\mathrm{MSE}} &= \mathrm{tr}\left\{\bar{\boldsymbol{\Sigma}}_{\delta_\phi}\mathbf{W}^\top\mathbf{W}\right\} \\ &= \mathrm{tr}\left\{\boldsymbol{\Sigma}_{\delta_\phi}\mathbf{W}^\top\mathbf{W}\right\} + \mathrm{tr}\left\{\delta\boldsymbol{\Sigma}_{\delta_\phi}\mathbf{W}^\top\mathbf{W}\right\},\end{aligned} \tag{4.46}$$

where the first equality derived from the property of $\mathrm{tr}\{\mathbf{ABC}\} = \mathrm{tr}\{\mathbf{BCA}\}$. The last equality is due to the definition of error

$$\bar{\boldsymbol{\Sigma}}_{\delta_\phi} = \boldsymbol{\Sigma}_{\delta_\phi} + \delta\boldsymbol{\Sigma}_{\delta_\phi}. \tag{4.47}$$

Taking expectation of (4.46), we obtain

$$\begin{aligned}\mathbb{E}\{\widehat{\mathrm{MSE}}\} &= \mathrm{tr}\left\{\boldsymbol{\Sigma}_{\delta_\phi}\mathbb{E}[\mathbf{W}^\top\mathbf{W}]\right\} + \mathbb{E}\left[\mathrm{tr}\{\delta\boldsymbol{\Sigma}_{\delta_\phi}\mathbf{W}^\top\mathbf{W}\}\right] \\ &= \mathrm{tr}\left\{\boldsymbol{\Sigma}_{\delta_\phi}\mathrm{Cov}(\mathbf{W}^\top)\right\} + \mathcal{O}\left(\frac{1}{\sqrt{T}}\right).\end{aligned} \tag{4.48}$$

Under asymptotic conditions, i.e. $T \to \infty$, the covariance $\bar{\boldsymbol{\Sigma}}_{\delta_\phi}$ converges. As the convergence rate, $\delta\boldsymbol{\Sigma}_{\delta_\phi}$ is proportional to $1/\sqrt{T}$. The detailed derivation can be found in [135].

4.7.2 Final Form of (4.21)

For ease of reference, we list some useful partial derivatives. Properties are not proved below can be found in [136].

$$\frac{\partial \det(\mathbf{X}^\top\mathbf{AX})}{\partial\mathbf{X}} = 2\det(\mathbf{X}^\top\mathbf{AX})\mathbf{X}^{-\top}, \tag{4.49}$$

$$\frac{\partial}{\partial\mathbf{X}}\mathrm{tr}\{\mathbf{XBX}^\top\mathbf{C}\} = \mathbf{C}^\top\mathbf{XB}^\top + \mathbf{CXB}. \tag{4.50}$$

Let $\mathbf{X}$ be square and invertible matrix, $\frac{\partial \det(\mathbf{X}^\top\mathbf{AX})}{\partial\mathbf{X}} = 2\det(\mathbf{X}^\top\mathbf{AX})\mathbf{X}^{-\top}$. Then

$$\frac{\partial \det(\mathbf{XAX}^\top)}{\partial\mathbf{X}} = 2\det(\mathbf{XAX}^\top)\mathbf{X}^{-\top}. \tag{4.51}$$

Proof. Starting from the left-hand side,

$$\begin{aligned}\frac{\partial \det(\mathbf{XAX}^\top)}{\partial\mathbf{X}} &= \frac{\partial[\det(\mathbf{X})\det(\mathbf{A})\det(\mathbf{X}^\top)]}{\partial\mathbf{X}} \\ &= \frac{\partial \det(\mathbf{X}^\top\mathbf{AX})}{\partial\mathbf{X}} \\ &= 2\det(\mathbf{X}^\top\mathbf{AX})\mathbf{X}^{-\top} \\ &= 2\det(\mathbf{X}^\top)\det(\mathbf{A})\det(\mathbf{X})\mathbf{X}^{-\top} \\ &= 2\det(\mathbf{X})\det(\mathbf{A})\det(\mathbf{X}^\top)\mathbf{X}^{-\top}\end{aligned}$$

$$= 2\det(\mathbf{X}\mathbf{A}\mathbf{X}^\top)\mathbf{X}^{-\top} \tag{4.52}$$

□

In (4.20), the derivative of the relative gradient $\nabla\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_s)$ with respect to the demixing matrix is derived

$$\frac{\partial\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_s)}{\partial\mathbf{W}} = \frac{1}{2}\left[\frac{\partial}{\partial\mathbf{W}}\mathrm{tr}\{\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi\mathbf{W}^\top\boldsymbol{\Sigma}_s^-\} - \frac{\partial}{\partial\mathbf{W}}\log\det(\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi\mathbf{W}^\top\boldsymbol{\Sigma}_s^-)\right], \tag{4.53}$$

for use in a matrix gradient based optimization algorithm. The derivative of the first term in the trace operator depends on the (4.50). The second derivative is readily computed by (4.51). Then we have

$$\begin{aligned}\frac{\partial\mathcal{J}(\mathbf{W}, \boldsymbol{\Sigma}_s)}{\partial\mathbf{W}} &= \frac{1}{2}\left[(\boldsymbol{\Sigma}_s^{-\top}\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi^\top + \boldsymbol{\Sigma}_s^-\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi) - \frac{1}{\det(\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi\mathbf{W}^\top\boldsymbol{\Sigma}_s^-)}\frac{\partial\det(\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi\mathbf{W}^\top\boldsymbol{\Sigma}_s^-)}{\partial\mathbf{W}}\right] \\ &= \frac{1}{2}\left[(\boldsymbol{\Sigma}_s^{-\top}\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi^\top + \boldsymbol{\Sigma}_s^-\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi) - \frac{2\det(\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi\mathbf{W}^\top\boldsymbol{\Sigma}_s^-)\mathbf{W}^{-\top}}{\det(\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi\mathbf{W}^\top\boldsymbol{\Sigma}_s^-)}\right] \\ &= \frac{1}{2}\left[\boldsymbol{\Sigma}_s^{-\top}\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi^\top + \boldsymbol{\Sigma}_s^-\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi - 2\mathbf{W}^{-\top}\right],\end{aligned} \tag{4.54}$$

In Addition, it is suggested in [130] to use natural gradient updates for faster convergence. The natural gradient is the gradient given in (4.20) postmultiplied by $\mathbf{W}^\top\mathbf{W}$ and is used to compute the following

$$\begin{aligned}-\frac{\partial\mathcal{J}}{\partial\mathbf{W}}\mathbf{W}^\top\mathbf{W} &= -\frac{1}{2}\left[\boldsymbol{\Sigma}_s^{-\top}\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi^\top + \boldsymbol{\Sigma}_s^-\mathbf{W}\bar{\boldsymbol{\Sigma}}_\phi - 2\mathbf{W}^{-\top}\right]\mathbf{W}^\top\mathbf{W} \\ &= -\frac{1}{2}\left[\boldsymbol{\Sigma}_s^{-\top}\bar{\boldsymbol{\Sigma}}_s^\top\mathbf{W} + \boldsymbol{\Sigma}_s^-\bar{\boldsymbol{\Sigma}}_s\mathbf{W} - 2\mathbf{W}\right] \\ &= \left[\mathbf{I} - \frac{1}{2}\boldsymbol{\Sigma}_s^-(\bar{\boldsymbol{\Sigma}}_s^\top + \bar{\boldsymbol{\Sigma}}_s)\right]\mathbf{W}.\end{aligned} \tag{4.55}$$

To substitute (4.55) into (4.20), the natural gradient used to update the demixing matrix could be given in (4.21).

4.7.3 Derivation of the Contrast Function

In this appendix, we prove the equality of (4.23). The desired identity is

$$\mathbb{E}\|\bar{\boldsymbol{\Sigma}}_{\delta_\phi} - \boldsymbol{\Sigma}_{\delta_\phi}\|^2 \leq \|\mathbb{E}[\bar{\boldsymbol{\Sigma}}_{\delta_\phi}] - \boldsymbol{\Sigma}_{\delta_\phi}\|^2 + \mathbb{E}\|\bar{\boldsymbol{\Sigma}}_{\delta_\phi} - \mathbb{E}[\bar{\boldsymbol{\Sigma}}_{\delta_\phi}]\|^2. \tag{4.56}$$

Proof. The performance error of an estimator $\bar{\boldsymbol{\Sigma}}_{\delta_\phi}$ with respect to an unknown parameter $\boldsymbol{\Sigma}_{\delta_\phi}$ is defined as

$$\begin{aligned}\mathbb{E}\|\bar{\boldsymbol{\Sigma}}_{\delta_\phi} - \boldsymbol{\Sigma}_{\delta_\phi}\|^2 &= \mathbb{E}\|\bar{\boldsymbol{\Sigma}}_{\delta_\phi} - \mathbb{E}[\bar{\boldsymbol{\Sigma}}_{\delta_\phi}] + \mathbb{E}[\bar{\boldsymbol{\Sigma}}_{\delta_\phi}] - \boldsymbol{\Sigma}_{\delta_\phi}\|^2 \\ &\leq \mathbb{E}\left\|\bar{\boldsymbol{\Sigma}}_{\delta_\phi} - \mathbb{E}[\bar{\boldsymbol{\Sigma}}_{\delta_\phi}]\right\|^2 + \mathbb{E}\left\|\mathbb{E}[\bar{\boldsymbol{\Sigma}}_{\delta_\phi}] - \boldsymbol{\Sigma}_{\delta_\phi}\right\|^2 \\ &\quad + \mathbb{E}\left\|2(\bar{\boldsymbol{\Sigma}}_{\delta_\phi} - \mathbb{E}[\bar{\boldsymbol{\Sigma}}_{\delta_\phi}])(\mathbb{E}[\bar{\boldsymbol{\Sigma}}_{\delta_\phi}] - \boldsymbol{\Sigma}_{\delta_\phi}])\right\|.\end{aligned} \tag{4.57}$$

The first equality just adds and subtracts the quantity $\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}]$ inside the norm. Then, the inequality is from the triangle inequality. Through observing the last equality, we know that both $\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}]$ and $\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi}$ are constants. Therefore, the only remaining the task with respect to the expectation is to compute $\bar{\mathbf{\Sigma}}_{\delta_\phi}$ profile, which leads us to have (4.56). □

4.7.4 A Closed-Form Expression for $\|\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi}\|^2$

For deriving the upper-bound of bias, the matrix norm in Definition 3 is applied to bound the spectral radius.

Proof. Starting from the expression of bias

$$\begin{aligned}
\left\|\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi}\right\|_2^2 &= \left\|((\omega_{ij}-1)\sigma_{ij})_{1\le i,j\le k}\right\|_2^2 \\
&\le \left\|((\omega_{ij}-1)\sigma_{ij})_{1\le i,j\le k}\right\|^2 \\
&= \left[\max_{1\le i\le k}\sum_j (\omega_{ij}-1)\sigma_{ij}\right]^2 \\
&\le \left[\max_{1\le i\le k}\sum_j |(\omega_{ij}-1)\sigma_{ij}|\right]^2 \\
&= \max_{1\le i\le k}\Bigg[\sum_{j:\frac{p}{2}\le|i-j|<p} |(\omega_{ij}-1)\sigma_{ij}| \\
&\quad + \sum_{j:p\le|i-j|} |(\omega_{ij}-1)\sigma_{ij}|\Bigg]^2 \\
&= \max_{1\le i\le k}\Bigg[\sum_{j:\frac{p}{2}\le|i-j|<p} \left|\left(1-\frac{2|i-j|}{p}\right)\sigma_{ij}\right| \\
&\quad + \sum_{j:p\le|i-j|} |\sigma_{ij}|\Bigg]^2, \qquad (4.58)
\end{aligned}$$

where the second equality used the definition of matrix norm as shown in Definition 3. The result of the fifth equality comes from the definition of the weight as defined in (4.29) that $\omega_{ij} = 1$ for $|i-j| < \frac{p}{2}$. By substituting (4.29) into the fifth equality of (4.58), we can obtain the sixth equality, which is an upper bound for $\|\mathbb{E}[\bar{\mathbf{\Sigma}}_{\delta_\phi}] - \mathbf{\Sigma}_{\delta_\phi}\|_2^2$. □

4.7.5 Proof of Lemma 1

Using the block matrices given in (4.34), we can set $\mathbf{S}^{(m)}$ as $\mathbf{S}^{(m)} = \sum_{l=1-m}^{k} \mathbf{M}_l^{(m)}$ when we assume k is divisible by m. The detailed derivation of Lemma 1 is given in the following.

Proof. Set $\boldsymbol{\delta}_l^{(m)} = \mathbf{M}_l^{(m)} - \mathbb{E}[\mathbf{M}_l^{(m)}]$, then we have

$$\begin{aligned}
\left\|\mathbf{S}^{(m)} - \mathbb{E}[\mathbf{S}^{(m)}]\right\| &= \left\|\sum_{l=1-m}^{k} \mathbf{M}_l^{(m)} - \mathbb{E}[\mathbf{M}_l^{(m)}]\right\| \\
&= \left\|\sum_{l=1-m}^{k} \boldsymbol{\delta}_l^{(m)}\right\| \\
&= \left\|\sum_{l=1}^{m} \sum_{-1\le j\le \frac{k}{m}} \boldsymbol{\delta}_{jm+l}^{(m)}\right\| \\
&\le \sum_{l=1}^{m} \left\|\sum_{-1\le j\le \frac{k}{m}} \boldsymbol{\delta}_{jm+l}^{(m)}\right\| \\
&\le m \max_{1\le l\le m} \left\|\sum_{-1\le j\le \frac{k}{m}} \boldsymbol{\delta}_{jm+l}^{(m)}\right\| \\
&\le m \max_{1-m\le l\le k} \left\|\boldsymbol{\delta}_l^{(m)}\right\| \\
&= m \max_{1\le l\le k} \left\|\boldsymbol{\delta}_l^{(m)}\right\|. \qquad (4.59)
\end{aligned}$$

Similarly, by setting $m_h = \frac{m}{2}$, we have

$$\left\|\mathbf{S}^{(m_h)} - \mathbb{E}[\mathbf{S}^{(m_h)}]\right\| \le m_h \max_{1\le l\le k} \left\|\boldsymbol{\delta}_l^{(m_h)}\right\|. \qquad (4.60)$$

Since $\boldsymbol{\delta}_l^{(m_h)}$ are all sub-blocks of $\boldsymbol{\delta}_l^{(m)}$, (4.60) can be written as

$$\left\|\mathbf{S}^{(m_h)} - \mathbb{E}[\mathbf{S}^{(m_h)}]\right\| \le m_h \max_{1\le l\le k} \left\|\boldsymbol{\delta}_l^{(m)}\right\|. \qquad (4.61)$$

Considering (4.59) and (4.61), we have

$$\begin{aligned}
\left\|\hat{\boldsymbol{\Sigma}}_{\delta_\phi}^{(m)} - \mathbb{E}[\hat{\boldsymbol{\Sigma}}_{\delta_\phi}^{(m)}]\right\| &= \frac{1}{m_h}\left\|\left(\mathbf{S}^{(m)} - \mathbb{E}[\mathbf{S}^{(m)}]\right) - \left(\mathbf{S}^{(m_h)} - \mathbb{E}[\mathbf{S}^{(m_h)}]\right)\right\| \\
&\le \frac{1}{m_h}\left[\left\|\mathbf{S}^{(m)} - \mathbb{E}[\mathbf{S}^{(m)}]\right\| + \left\|\mathbf{S}^{(m_h)} - \mathbb{E}[\mathbf{S}^{(m_h)}]\right\|\right] \\
&\le \frac{1}{m_h}\left[m \max_{1\le l\le k} \|\boldsymbol{\delta}_l^{(m)}\| + m_h \max_{1\le l\le k} \|\boldsymbol{\delta}_l^{(m)}\|\right] \\
&= 3 \max_{1\le l\le k} \|\boldsymbol{\delta}_l^{(m)}\| \\
&= 3\mathcal{N}_l^{(m)}. \qquad (4.62)
\end{aligned}$$

Then Lemma 1 immediately follows from (4.62). □

4.7.6 Proof of Lemma 2

Proof. According to (4.37), we have

$$\mathbb{P}\{\mathbf{v}^\top \left(\boldsymbol{\phi}(\mathbf{x}_i) - \mathbb{E}[\boldsymbol{\phi}(\mathbf{x}_i)]\right) \left(\boldsymbol{\phi}(\mathbf{x}_i) - \mathbb{E}[\boldsymbol{\phi}(\mathbf{x}_i)]\right)^\top \leq \exp(-x\rho/2) \tag{4.63}$$

Then, there is a constant $\rho_1 > 0$ such that

$$\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} \mathbf{v}^\top \left[(\boldsymbol{\phi}(\mathbf{x}_i) - \mathbb{E}[\boldsymbol{\phi}(\mathbf{x}_i)])(\boldsymbol{\phi}(\mathbf{x}_i) - \mathbb{E}[\boldsymbol{\phi}(\mathbf{x}_i)])^\top - \Sigma\right]\mathbf{v}\right| > x\right\} \leq \exp(-nx^2\rho_1/2) \tag{4.64}$$

Then we have

$$\begin{aligned}
\mathbb{P}\left\{\max_{1\leq l\leq p-m+1} \left\|\mathbf{M}_k^{(m)} - \mathbb{E}[\mathbf{M}_k^{(m)}]\right\| > x\right\} &\leq \max_{1\leq l\leq p-m+1} \mathbb{P}\left\{\left\|\mathbf{M}_k^{(m)} - \mathbb{E}[\mathbf{M}_k^{(m)}]\right\| > x\right\} \\
&\leq 2k5^m \sup_{v_j,l} \mathbb{P}\left\{\left|\mathbf{v}_j^\top \left(\mathbf{M}_k^{(m)} - \mathbb{E}[\mathbf{M}_k^{(m)}]\right)\mathbf{v}_j\right| > x\right\} \\
&\leq 2k5^m \exp(-nx^2\rho_1/2).
\end{aligned} \tag{4.65}$$

From the above Equation and Lemma 1, we have the bounded variance as

$$\begin{aligned}
\mathbb{E}\|\hat{\boldsymbol{\Sigma}}_{\delta_\phi}^{(m)} - \mathbb{E}[\hat{\boldsymbol{\Sigma}}_{\delta_\phi}^{(m)}]\|^2 &\leq 9\mathbb{E}[\mathcal{N}_l^{(m)}]^2 \\
&= 9\mathbb{E}[\mathcal{N}_l^{(m)}]^2 \left[\mathbb{P}(\mathcal{N}_l^{(m)} \leq x) + \mathbb{P}(\mathcal{N}_l^{(m)} > x)\right] \\
&\leq 9\left[x^2 + \mathbb{E}[\mathcal{N}_l^{(m)}]^2 \mathbb{P}(\mathcal{N}_l^{(m)} > x)\right].
\end{aligned} \tag{4.66}$$

Since $\|\mathbb{E}[\hat{\boldsymbol{\Sigma}}_{\delta_\phi}^{(m)}]\| \leq \boldsymbol{\Sigma}_{\delta_\phi}^{(m)}$, then $\mathcal{N}_l^{(m)}$ is bounded by a constant $\mathcal{C}$. Then (4.66) can be written as

$$\begin{aligned}
\mathbb{E}\|\hat{\boldsymbol{\Sigma}}_{\delta_\phi}^{(m)} - \mathbb{E}[\hat{\boldsymbol{\Sigma}}_{\delta_\phi}^{(m)}]\|^2 &\leq 9\left[x^2 + \mathcal{C}^2 \mathbb{P}(\mathcal{N}_l^{(m)} > x)\right] \\
&\leq 9\left[x^2 + \mathcal{C}^2 2k5^m \exp(-nx^2\rho_1)\right].
\end{aligned} \tag{4.67}$$

Since x is bounded as $0 < x < \rho_1$, then we have $\mathbb{E}\|\hat{\boldsymbol{\Sigma}}_{\delta_\phi}^{(m)} - \mathbb{E}[\hat{\boldsymbol{\Sigma}}_{\delta_\phi}^{(m)}]\|^2$ is bounded by a constant.

□

Chapter 5

Kernelized Feature Subspace-based Underdetermined BSS

In general, most blind source separation (BSS) algorithms assume that the number of sources is less than that of sensors, denoted as overdetermined BSS. However, in practice, this assumption is difficult to be satisfied since the number of sources is unknown. In this Chapter, we introduce a model that relies on a Kernelized multi-subspace and the sparse representation in the time-frequency (TF) domain to solve the underdetermined BSS problem. The overview of some relative works and the fundamental problems are given in Chapter 5.1. Chapter 5.2 is the preliminary that reviews the consents of convex geometry, Kernel theorem first. Then, the nonlinear mixture model is introduced for further study. Chapter 5.3 introduces some conditions necessary for the separation of nonstationary sources in the TF domain. Chapter 5.4 describes our proposed separation approach that relies on multi-subspaces representation and sparse representation in the TF domain. Chapter 5.5 shows the experimental settings and results. Conclusions are reported in Chapter 5.6.

5.1 Introduction

Various attempts [137, 52, 138] on underdetermined BSS (UBSS) have been proposed that consider the scenario, where the number of sensors is less than that of sources. Since the mixing matrix is irreversible in this case, the recovered sources also need to be estimated even though the mixing matrix has been known. To solve this problem, a well-known framework has been proposed by exploiting the sparseness of the sources in the representation domain, such as wavelet packet transform [139] or short-time Fourier transform (STFT) [140]. For instance, the degenerate unmixing estimation technique (DUET) was proposed in [141]. The approach exploits the ratio of TF transforms of the observed signals to recover the source signals. Yilmaz et al. [142] assumed that the sources are disjoint in the TF domain. These methods work on the assumption that there exists at most one active source at any point in the TF domain. This implies that the separation performance will degrade as the number of the TF disjoints points being increased. To relax this constraint, [53, 143] proposed a scenario that allows the sources to be non-disjoint in the TF domain, however, the number of the sources that coexist at any TF point is less than that of the mixtures [53].

In the above methods, the mixing process is considered to be linear only. In fact, however, the assumption is restrictive and easy to be violated in the real-world applications [144], such as communication [145, 146], speech or audio processing [147], and biomedical engineering [148]. The problem for the nonlinear BSS is intractable solely based on the assumption that the sources are statistically independent. e.x., if x and y are two independent random variables, then $f(x)$ and $g(y)$ are also independent for any f and g [149]. Therefore, the solutions are highly non-unique without any further constraints for the space of nonlinear mixing function [29].

Efforts on exploiting such further constraints in the nonlinear domain have involved, such as extracting unknown nonlinearities upon unknown parameters [25], approximating a nonlinear function whose inverse function can be constrained well on the estimator of a priori neural network [145, 150]. Another popular approach consists in using kernel so as to implicitly map the data via kernel trick. The main advantage of this approach is that the estimation of the parameters in the model is actually independent of the number of channels. Formally, the data are mapped into $\mathcal{H}$

using $\phi : \mathcal{X} \rightarrow \mathcal{H}, \mathbf{x} \rightarrow \phi(\mathbf{x})$ so as to extract the nonlinearity. To avoid working on the high-dimensional space $\mathcal{H}$, one tries in the feature space in which the dot product can be calculated by $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$, which is called as kernel trick.

Typically, Harmeling and Martinez [44, 151] exploit the temporal information of sources for separation, and do not enforce mutual independence of outputs. This method produces successful results in many experiments. However, a problem is that the cost of storing and evaluating the model is proportional to the number of data points [46]. Moreover, this method may fail if some sources lack specific time structures. [126, 152] provides a good approximation of the value attained by the nonlinear mixing. Relying on such spaces spanned by a set of vanishing polynomial, the data implicitly mapped into high-dimensional space, and the effective subspace is extracted. It allowed us to solve a nonlinear problem linearly. But, unfortunately, the approach can not be used for the underdetermined case.

In this chapter, we propose a multi-subspace representation based separation approach that tackles the scenario of the nonlinear and underdetermined mixture. The separation system is constructed using the kernel methods with a multi-subspace structure. To obtain a set of basis so as to the spanned subspace could be orthonormal in the theoretical support, we propose to use the geometric vertices of data. Then we solve a linear problem by exploiting the technique of sparse coding. The coefficient matrix is adjusted by minimizing the loss function.

We first consider a model related to the input space $\mathbf{x} \in \mathbb{R}^N$ by a kernel mapping with multi-subspace structure. The effective number of basis denoted by k, provides the smallest construction error in the nonlinear approximation. One of the keys in that algorithm is to find a set of orthogonal basis to study the parameterized signals in multiple feature spaces. Some techniques [42, 43, 44] can help that are roughly analogous. Either random sampling or k-means clustering is considered to obtain some vectors, which is expected to be independent. However, the method may not be appropriate for mixture data. We attempt to use the geometric vertices of the convex hull as the basis, which parameterizes the multi-subspace that contains the reduced vectors in the feature space. Relying on a set of an orthonormal basis, the spanned subspaces can represent the nonlinearity of mixing function in the minimum number.

Another contribution is to derive the coefficient matrix by solving the loss function on the coding coefficient vector. Once such subspaces are built, by allowing multiple sources to be presented at any point in the TF domain, we can figure out the target matrix in a sparse mixture TF vectors with less computational cost. Finally, using this coefficient matrix, the original sources in underdetermined scenarios can be estimated.

5.2 Preliminary and System Model

In the following, a brief review of some concepts on convex geometry and Kernel method will be given for ease of later use.

5.2.1 Convex Geometry

The Definition 9 of convex hull [153] for a set of vectors $\{\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T)\}$ will be given in the following.

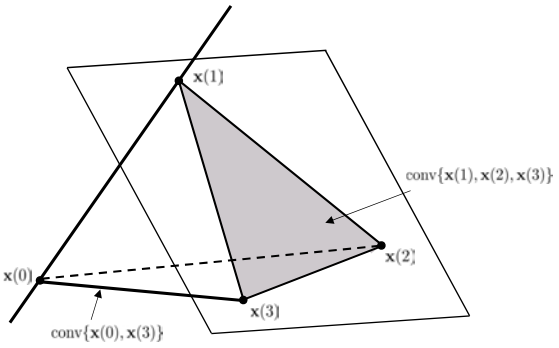


Figure 5.1: A Graphical Illustration for the Convex Geometry Concepts. The line segment connecting $\mathbf{x}(0)$ and $\mathbf{x}(3)$ is the convex hull of $\{\mathbf{x}(0), \mathbf{x}(3)\}$, which is denoted by $\mathrm{conv}\{\mathbf{x}(0), \mathbf{x}(3)\}$. The shaded triangle is the convex hull of $\{\mathbf{x}(1), \mathbf{x}(2), \mathbf{x}(3)\}$, i.e., $\mathrm{conv}\{\mathbf{x}(1), \mathbf{x}(2), \mathbf{x}(3)\}$.

Definition 9. *Given a set of vectors $\mathcal{X} = \{\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T)\}$. The convex hull of the finite nonempty set $\mathcal{X} \subseteq \mathbb{R}^N$ gives the form*

$$\mathrm{conv}\{\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T)\} \stackrel{\triangle}{=} \left\{ \sum_{i=1}^{T} \lambda_i \mathbf{x}(i) \middle| \boldsymbol{\lambda} \in \mathbb{R}_+^T, \mathbf{1}_T^\top \boldsymbol{\lambda} = 1 \right\},$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \cdots, \lambda_T]^\top$ is any non-negative vector . □

In the above equation, $\mathrm{conv}\{\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T)\}$ is called as a $(T-1)$-dimensional simplex with T vertices $\{\mathbf{x}(1), \{\mathbf{x}(2), \cdots, \mathbf{x}(T)\}$ if and only if $\{\mathbf{x}(1), \{\mathbf{x}(2) \cdots, \mathbf{x}(T)\}$ is affinely independent, or equivalently. Furthermore, if $\{\mathbf{x}(1) - \mathbf{x}(T), \mathbf{x}(2) - \mathbf{x}(T), \cdots, \mathbf{x}(T-1) - \mathbf{x}(T)\}$ is linearly independent that is called a simplest simplex in $\mathbb{R}^N$ [154]. As see in the Fig. 5.1, a triangle is a 2-dimensional simplest simplex in $\mathbb{R}^2$, and a tetrahedron is a 3-dimensional simplest simplex in $\mathbb{R}^3$.

According to the N-FINDR criterion [54], the approach finds the endmembers' convex hull that in fact of extracting the data-enclosing simplex with the maximum volume [155]. That can be given by solving the maximization problem

$$\max_{\mathbf{p}(i) \in \mathbb{R}^{M-1}, \forall i} \mathcal{V}(\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k))$$
$$\text{s.t.} \quad \mathbf{x}(t) \in \mathrm{conv}\{\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k)\}, \ \forall t$$

where $\mathcal{V}(\cdot)$ denotes the volume of the simplex $\mathrm{conv}\{\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k)\} \subseteq \mathbb{R}^{M-1}$.

The above theory is introduced for the theoretical support in our further work, where the geometric vertices can establish a set of orthogonal basis so that the spanned multiple subspaces can represent the nonlinearity in the minimum number.

5.2.2 Nonlinear Mixture Model

Consider a nonlinear, instantaneous and invertible mixing system with M inputs and N outputs

$$\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t)), \tag{5.1}$$

for $t = 1, 2, \cdots, T$, where $\mathbf{s}(t) = [s_1(t), s_2(t), \cdots, s_M(t)]^\top$ is the original sources of M statistically independent vectors. The superscript $[\cdot]^\top$ denotes the transpose operator. $s_i(t)$ denotes the original source of the i-th signal at t time index. The mixing function $\mathcal{F}$ transform the $\mathbf{s}(t)$ from $\mathbb{R}^M$ to $\mathbb{R}^N$, i.e., the observations $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_N(t)]^\top$ are N-dimensional mixture vectors.

The general idea of performing is to design a separation function $\mathcal{G} : \mathbb{R}^N \to \mathbb{R}^M$ such that

$$\hat{\mathbf{s}}(t) = \mathcal{G}(\mathbf{x}(t)), \tag{5.2}$$

where the recovered sources $\hat{\mathbf{s}}$ are statistically independent. One has been given in [11], where the nonlinear mixtures of independent variables are still independent. However, the statistical independence of estimated sources is no longer a sufficient constraint for demixing function, without additional prior knowledge on the mixing process [29]. To form a mapping function with multi-subspace structure, we consider the Kernel theorem and its feature space.

5.2.3 Kernel and Feature Space

The key point is how to generate a mapping function that can achieve the approximation of the inverse operator of (5.1). In [44], the kernelization method was introduced by mapping the data $\mathbf{x}(t)$ implicitly into the kernel feature space $\mathcal{H}$ with the kernel function $\mathcal{K} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$. The basic definitions are introduced at first.

Definition 10. *Let $\mathcal{X}$ be a nonempty set. The symmetric function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called as a positive definite kernel, if*

$$\sum_{i,j=1}^{N} c_i c_j \mathcal{K}(\mathbf{x}(i), \mathbf{x}(j)) \geq 0, \tag{5.3}$$

holds for any $\mathbf{x}(i) \in \mathcal{X}$ and $c_1, c_2, \cdots, c_N \in \mathbb{R}$. □

One can easily deduce from Definition 10 that the positive definite kernel transforms data into kernel feature space, which can be simply calculated by matrices of kernel built on the sample of points as

$$\langle \boldsymbol{\phi}(\mathbf{x}(i)), \boldsymbol{\phi}(\mathbf{x}(j)) \rangle = \mathcal{K}(\mathbf{x}(i), \mathbf{x}(j)), \tag{5.4}$$

where $i, j = 1, 2, \cdots, T$ and $\langle \cdot, \cdot \rangle$ is the inner product. $\boldsymbol{\phi}(\mathbf{x})$ is the Hilbert mapping function. Using the kernel trick, the inner product of two feature mappings in the Hilbert space can be computed by a kernel function in the original space. The computational complexity can be controlled within a linear range.

This would first define a direction $\mathbf{W} \in \mathcal{H}$ that enables us to parameterize the data by

$$\mathbf{W} = \boldsymbol{\Phi}_{\mathbf{x}} \boldsymbol{\alpha} = \sum_{j=1}^{T} \alpha_j \boldsymbol{\phi}(\mathbf{x}(j)) \in \mathcal{H}, \tag{5.5}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_T]^\top$ is a parameter vector. $\boldsymbol{\Phi}_{\mathbf{x}}$ is the matrix with the column vectors $[\boldsymbol{\phi}(\mathbf{x}(1)), \boldsymbol{\phi}(\mathbf{x}(2)), \cdots, \boldsymbol{\phi}(\mathbf{x}(T))]^\top$. Using the kernel trick of (5.4), the demixing process in the feature space is given by

$$\hat{\mathbf{s}}(t) = \mathbf{W}^\top \boldsymbol{\Phi}(\mathbf{x}(t)) = \boldsymbol{\alpha}^\top \boldsymbol{\Phi}_{\mathbf{x}}^\top \boldsymbol{\phi}(\mathbf{x}(t)) = \sum_{j=1}^{T} \alpha_j \mathcal{K}(\mathbf{x}(j), \mathbf{x}(t)). \tag{5.6}$$

The main advantage of Kernel mapping is that the number of parameters to estimate in the model is actually independent of the number of channels. However, without extra constraints, generating a unique mapping function is intractable.

This chapter proposes a multi-subspace representation based on Kernel spaces to tackle the ill-posed with a few assumptions. The k multiple subspaces produce k outputs, and we propose the way to select n outputs as the estimator of the original sources.

5.3 Linear TF-UBSS Approach

We first review the TF domain based underdetermined BSS (UBSS) method that was presented by [53] and later proposes a multi-layer representation based nonlinear TF-UBSS algorithm. The discrete-time short-time Fourier transform (STFT) is given by

$$\mathcal{D}_{s_i}(\tau, \omega) \triangleq \sum_{t=-\infty}^{\infty} s_i(t) h(t-\tau) e^{-j\omega t}, \tag{5.7}$$

at frame τ and frequency bin ω, where $h(t)$ is a window function. Using STFT of (6.4), the linear BSS can be transformed into the TF domain

$$\mathcal{D}_{\mathbf{x}}(t, \omega) = \mathbf{A} \mathcal{D}_{\mathbf{s}}(t, \omega), \tag{5.8}$$

where $\mathcal{D}_{\mathbf{x}}(t, \omega) = [\mathcal{D}_{x_1}(t, \omega), \mathcal{D}_{x_2}(t, \omega), \cdots, \mathcal{D}_{x_N}(t, \omega)]^\top$ is the mixture signals in the TF domain and $\mathcal{D}_{\mathbf{s}}(t, \omega) = [\mathcal{D}_{s_1}(t, \omega), \mathcal{D}_{s_2}(t, \omega), \cdots, \mathcal{D}_{s_M}(t, \omega)]^\top$ is the STFT vector of the source signals. $\mathcal{D}_{s_i}(t, \omega)$ is the i-th source signal in the ω-th frequency bin at t time index.

Assumption 1. *For each source signal $\mathbf{s}_i$, its STFT transformation is denoted as $\mathcal{D}_{\mathbf{s}_i}$ in the TF domain. There are some TF points, where only $\mathbf{s}_i$ is dominant, i.e., $|\mathcal{D}_{\mathbf{s}_i}(t, \omega)| \gg |\mathcal{D}_{\mathbf{s}_j}(t, \omega)|$ for $\forall j \neq i$.* □

The assumption implies that all sources are disjoint in the TF domain, i.e., there is only one source that is active. Then, (6.17) can be rewritten as

$$\mathcal{D}_{\mathbf{x}}(t_a, \omega_a) = \mathbf{a}_i \mathcal{D}_{\mathbf{s}_i}(t_a, \omega_a), \tag{5.9}$$

where the subscript a indicates any one of the sources is active in the TF domain.

The noise thresholding procedure proposed by [142] is used to keep those points having sufficient energy, which is referred to as auto-source points. The procedure is performed for each time-slice of the TF representation, by applying a criterion for all the frequency points belonging

to this time-slice

$$\text{if} \quad \frac{\|\mathcal{D}_{\mathbf{x}}(t_a,\omega_a)\|}{\max_{\omega}\{\|\mathcal{D}_{\mathbf{x}}(t_a,\omega)\|\}} > \epsilon, \quad \text{then keep } (t_a,\omega_a), \tag{5.10}$$

where ϵ is a small threshold, e.x., the threshold $\epsilon = 0.05$ is given in [53]. Then, the set of all selected points Ω is expressed by $\Omega = \bigcup_{i=1}^{n} \Omega_i$, where Ω_i is the TF support of the source $s_i(t)$.

To estimate the mixing vectors $\mathbf{a}_i$, the clustering algorithm is performed on the assumption in [53] that the highest densities occur around the vectors $\mathbf{a}_i$. Thus, the average values over the samples of each cluster are defined as the mixing vectors

$$\hat{\mathbf{a}}_i = \frac{1}{|C_i|} \sum_{(t,\omega)\in\Omega_i} \frac{\mathcal{D}_{\mathbf{x}}(t,\omega)}{\|\mathcal{D}_{\mathbf{x}}(t,\omega)\|}, \tag{5.11}$$

where $|C_i|$ is the number of vectors included in the same cluster.

Finally, each source in the TF domain can be estimated by

$$\hat{\mathcal{D}}_{s_i}(t,\omega) = \begin{cases} \hat{\mathbf{a}}_i^{\dagger}\mathcal{S}_{\mathbf{x}}(t,\omega), & \forall (t,\omega) \in \Omega_i, \\ 0, & \text{otherwise}, \end{cases} \tag{5.12}$$

where the superscript $[\cdot]^{\dagger}$ denotes the pseudo-inverse operator. The source estimator $\hat{s}_i(t)$ is then obtained by transforming $\hat{\mathcal{D}}_{s_i}(t,\omega)$ into the time domain using the inverse STFT.

5.4 Multi-Subspace Representation based Nonlinear TF-UBSS Approach

The TF-UBSS method relies on the assumption that the sources were mixed linearly, which has led to the recovered structure in (5.12). However, for the nonlinear blind source separation, the solutions are non-unique [29] without any extra constraints for the mixing process. In this chapter, we propose a multi-subspace representation to construct the nonlinear variants by mapping the data implicitly in some kernel feature spaces. If one of the subspaces can match the nonlinearity of the mixing functions, the nonlinear problem can be broken down into the version of the linear case.

5.4.1 Choosing Vectors for Basis

To extract a vector that formed a matrix with full column rank, we use the N-FINDR algorithm, which was originally developed by Winter in [54]. The approach finds a set of vertices in fact of extracting a vector of data space that defined the largest volume.

Definition 11. *Let $\mathcal{X} = \{\mathbf{x}(i)\}_{i=1}^{T}$ be a set of sample vectors. The convex hull of the finite nonempty set $\mathcal{X} \subseteq \mathbb{R}^d$ gives the form*

$$\text{conv}(\{\mathbf{x}(1),\mathbf{x}(2),\cdots,\mathbf{x}(T)\}) \stackrel{\triangle}{=} \left\{ \sum_{i=1}^{T} \lambda_i \mathbf{x}(i) \big| \lambda_i \geq 0, \sum_i \lambda_i = 1 \right\}, \tag{5.13}$$

□

Proposition 1. *Let* $\{\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k)\}$ *be a subset of vectors in the convex hull* $\mathcal{X} = \{\mathbf{x}(i)\}_{i=1}^{T}$. *For* $k \ll T$*, if the vectors* $\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k)$ *are the vertices of* $\mathcal{X}$*, then we have*

$$\operatorname{conv}(\{\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T)\}) \subseteq \operatorname{conv}(\{\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k)\}). \tag{5.14}$$

□

Proof. Without loss of generality, $\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(k)$ are the vertices of $\mathcal{P} := \operatorname{conv}(\{\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T)\})$, which are expressed as $\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k)$. For any $i > k$, if $\mathbf{x}(i)$ is not a vertex of $\mathcal{P}$, then $\mathbf{x}(i)$ can be expressed by a linear combination $\mathbf{x}(i) = \sum_{j=1}^{k} \lambda_j \mathbf{p}(j)$. Thus, for any sample $\mathbf{x} \in \mathcal{P}$, we have

$$\begin{aligned}
\mathbf{x} &= \sum_{i=1}^{T} \mu_i \mathbf{x}(i) = \sum_{i=1}^{k} \mu_i \mathbf{p}(i) + \sum_{i=k+1}^{T} \mu_i \mathbf{x}_i \\
&= \sum_{i=1}^{k} \mu_i \mathbf{p}(i) + \sum_{i=k+1}^{T} \mu_i \sum_{j=1}^{k} \lambda_j \mathbf{p}(j) \\
&= \sum_{i=1}^{k} \left(\mu_i + \lambda_i \sum_{j=k+1}^{T} \mu_j \right) \mathbf{p}(i).
\end{aligned} \tag{5.15}$$

Since $\sum_{i=1}^{k}(\mu_i + \lambda_i \sum_{j=k+1}^{T} \mu_j) = 1$, we conclude that $\mathbf{x} \in \operatorname{conv}(\{\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(T-1)\}) \subseteq \operatorname{conv}(\{\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k)\})$. □

Proposition 1 implies that the volume simplex formed by the vertices is larger than or equal to any other volume defined by any other combination of elements. Thus, the vertices can be extracted in fact of finding a vector of data space that formed the maximum volume. The approach can be briefly described in the following implementation.

For a vertex simplex composed of k vectors $\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k)$, its volume $\mathcal{V}(\mathbf{P}) = \mathcal{V}(\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k))$ is defined by

$$\mathcal{V}(\mathbf{P}) \triangleq \frac{\left| \det \begin{bmatrix} 1 & \cdots & 1 \\ \mathbf{p}(1) & \cdots & \mathbf{p}(k) \end{bmatrix} \right|}{(k-1)!}. \tag{5.16}$$

Find a set of k vectors in the data, denoted by $\mathbf{P}^* = [\mathbf{p}^*(1), \mathbf{p}^*(2), \cdots, \mathbf{p}^*(k)]$, that forms a k-vertex simplex to yield the maximum value of (5.16), which is given by

$$\{\mathbf{p}^*(i_1), \mathbf{p}^*(i_2), \cdots, \mathbf{p}^*(i_k)\} = \arg \max_{\mathbf{p}(i_1), \mathbf{p}(i_2), \cdots, \mathbf{p}(i_k)} \mathcal{V}(\mathbf{P}). \tag{5.17}$$

Thus, the desired set of independent vectors $\{\mathbf{p}^*(i_1), \mathbf{p}^*(i_2), \cdots, \mathbf{p}^*(i_k)\}$ are found. Assume that the dimension of vector $\mathbf{p}^*$ is larger that the number of vector k, then the columns of the matrix being linearly independent. For further work, a set of orthonormal subspaces produced by these k vectors can represent the nonlinearity or distortion caused by the mixing functions using the reduced data.

5.4.2 Constructing a Multi-Subspace Representation

Given the observation data $\mathbf{x}(t) \in \mathbb{R}^N$, for all $t = 1, \cdots, T$ that are assumed to be generated by the nonlinear mixture functions. To make the nonlinear problem linearly separable, the idea is to fulfill a certain condition that induces a mapping $\Phi : \mathbb{R}^N \to \mathcal{H}$ in the feature space. Therefore, we attempt to find some mapping functions, which are used to capture the varieties of nonlinearity or distortion.

To describe the nonlinearity efficiently in a feature space, we use a subset from $\{\mathbf{x}(t)\}_{t=1}^T \in \mathbb{R}^N$, denoted as $\mathbf{p}(1), \mathbf{p}(2), \cdots, \mathbf{p}(k) \in \mathbb{R}^N$ to generate a set of basis in $\mathcal{H}$. Since the data points belonging to the subset is expected to be mutually independent in the feature space, we use the k center points of clusters to form the subset $\{\mathbf{p}(i)\}_{i=1}^k$. Thus, we can define an orthonormal basis by using the empirical kernel map

$$\mathbf{\Xi} := \mathbf{\Phi_p} \langle \mathbf{\Phi_p}, \mathbf{\Phi_p} \rangle^{-\frac{1}{2}}, \tag{5.18}$$

where $\mathbf{\Phi_p} = [\Phi(\mathbf{p}_1), \Phi(\mathbf{p}_2), \cdots, \Phi(\mathbf{p}_k)]$ is the mapping of data points in the feature space.

By defining the basis that allows us to parameterize such subspace, the observed signals are mapped in the feature space with the coefficient matrix from a parameter space.

$$\begin{aligned} \mathbf{\Psi}(\mathbf{x}(t)) &= \mathbf{\Xi}^\top \mathbf{\Phi}(\mathbf{x}(t)) = \langle \mathbf{\Phi_p}, \mathbf{\Phi_p} \rangle^{-\frac{1}{2}} \langle \mathbf{\Phi_p}, \mathbf{\Phi}(\mathbf{x}(t)) \rangle \\ &= \begin{bmatrix} \mathcal{K}(\mathbf{p}(1), \mathbf{p}(1)) & \cdots & \mathcal{K}(\mathbf{p}(1), \mathbf{p}(k)) \\ \vdots & & \vdots \\ \mathcal{K}(\mathbf{p}(k), \mathbf{p}(1)) & \cdots & \mathcal{K}(\mathbf{p}(k), \mathbf{p}(k)) \end{bmatrix}^{\frac{1}{2}} \begin{bmatrix} \mathcal{K}(\mathbf{p}(1), \mathbf{x}[t]) \\ \vdots \\ \mathcal{K}(\mathbf{p}(k), \mathbf{x}[t]) \end{bmatrix}, \end{aligned} \tag{5.19}$$

where $\mathcal{K}(\mathbf{p}(i), \mathbf{p}(j))_{i,j}^{-\frac{1}{2}}$ is an invertible real valued matrix. Due to the $\mathbf{\Phi_p}$ constructed by a subset, the computational complexity of the projection function in (5.19) is reduced to $\mathcal{O}(k^2 N) + \mathcal{O}(kNT) + \mathcal{O}(k^2 T)$ from original $\mathcal{O}(T^2 N) + \mathcal{O}(NT^2) + \mathcal{O}(T^3)$, where $T \gg k$.

Thus, the demixing process can be defined in the feature space as

$$\hat{\mathbf{s}}(t) = \mathbf{W}^\dagger \mathbf{\Psi}(\mathbf{x}(t)). \tag{5.20}$$

The above equation implies that the nonlinear problem can be linearly separable in the feature space.

5.4.3 Coefficient Matrix Identification

Relying on the linear relation of (5.20), we have the corresponding representation by using STFT,

$$\mathcal{D}_{\mathbf{\Psi}}(t, \omega) = \tilde{\mathbf{W}} \hat{\mathcal{D}}_{\mathbf{s}_i}(t, \omega). \tag{5.21}$$

Based on Assumptions 1, we know that there exists only one estimated source $\hat{\mathbf{s}}_i$ being active on the TF point (t, ω). Then, we have

$$\mathcal{D}_{\mathbf{\Psi}}(t, \omega) = \hat{\mathcal{D}}_{\mathbf{s}_i}(t, \omega) \tilde{\mathbf{W}}_i, \tag{5.22}$$

where the TF feature matrix $\mathcal{D}_{\mathbf{\Psi}}(t,\omega)$ can be represented by the i-th column vector $\tilde{\mathbf{W}}_i$ up to a multiplicative coefficient $\mathcal{D}_{\hat{\mathbf{s}}_i}(t,\omega)$. This implies that the target matrix $\tilde{\mathbf{W}}_i$ can be a linear combination of a few numbers of sample points from the matrix $\mathcal{D}_{\mathbf{\Psi}}(t,\omega)$ with the coefficient $\hat{\mathcal{D}}_{\mathbf{s}_i}(t,\omega)$.

Thus, estimating a column vector of the coefficient matrix $\tilde{\mathbf{W}}_i$ can be achieved by finding the solution of a sparse representation $\mathcal{D}_{\mathbf{\Psi}}(t,\omega)$ with low-dimensional subspace. To remove the effect of noise, we use the criterion for all the frequency points belonging to this time-slice

$$\text{if} \quad \frac{\|\mathcal{D}_{\mathbf{\Psi}}(t_p,\omega_k)\|}{\max_{\omega}\{\|\mathcal{D}_{\mathbf{\Psi}}(t_p,\omega)\|\}} > \epsilon, \quad \text{then keep } (t_p,\omega_k), \tag{5.23}$$

where ϵ is a small threshold, e.x., the threshold $\epsilon = 0.05$ is given in [53].

We next formulate the problem of (6.18) by using a sparse direction for TF representation of the mixture TF matrix $\mathcal{D}_{\mathbf{\Psi}}(t,\omega)$. Let $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_L$ be the reshaped vector of all the mixture TF matrix $\mathcal{D}_{\mathbf{\Psi}}$, and L is the number of TF points (t,ω). We can define a one row vector $\mathcal{D}_{\mathbf{\Pi}} \triangleq [\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_L]$ that is row-wise stacked together to be generated by the mixture TF matrix $\mathcal{D}_{\mathbf{\Psi}}$ at all (t,ω).

The further solution of (6.19) is the sparse representation of the TF feature vector $\mathcal{D}_{\mathbf{\Pi}}$, that will later construct the estimation of the coefficient matrix in the TF domain.

$$\mathcal{J}(\mathbf{c}_i,\eta) = \frac{1}{2}\|\boldsymbol{\pi}_i - \mathcal{D}_{\mathbf{\Pi}}\mathbf{c}_i\|_2^2 + \eta\|\mathbf{c}_i\|_1, \quad \text{s.t.,}\ \mathbf{c}_{ii} = 0, \tag{5.24}$$

where $\eta > 0$ is a scalar parameter to balance the trade-off between the sparsity and reconstruction error. $\mathbf{c}_i$ is the corresponding sparse coefficient for $\boldsymbol{\pi}_i$. The maximum value in $\mathbf{c}_i$ indicates the estimated element of $\mathbf{W}$ that is corresponding to $\mathcal{D}_{\mathbf{\Pi}}$. Once a sparse coding problem is built, the solution can be obtained by solving the convex optimization problem. Here, we use l_1-Homotopy method in [156] to calculate the redundant dictionary $\mathbf{c}_i$ of (6.19). The procedure obtains a sparse solution with $\mathcal{O}(q^3 + L)$ orders, where q is the number of non-zero elements.

5.4.4 Source Recovery

Since the mixing matrix is not irreversible in the underdetermined BSS [157], the recovered sources also need to be estimated even though the mixing matrix has been known. To obtain a sparse TF representation of the recovered sources, we use the process proposed by [52] with the definition of sub-matrix $\boldsymbol{\mathcal{W}}$ on the following assumption.

Assumption 2. *At most $N-1$ sources among M sources are active at each TF point for $M > N$ [53].* □

Definition 12. *Given a matrix $\mathbf{W}$ of size $N \times M$, for any sub-matrices $\boldsymbol{\mathcal{W}}_i$ composed of size $N \times (N-1)$, there are $\binom{M}{N-1}$ possible combinations included in the set $\boldsymbol{\mathcal{W}}$, that is*

$$\boldsymbol{\mathcal{W}} = \{\boldsymbol{\mathcal{W}}_i | \boldsymbol{\mathcal{W}}_i = [\boldsymbol{w}_{\lambda_1}, \boldsymbol{w}_{\lambda_2}, \cdots, \boldsymbol{w}_{\lambda_{N-1}}]\}. \tag{5.25}$$

□

Assumption 2 indicates the number of columns of the sub-matrix $\boldsymbol{\mathcal{W}}_i$ to be derived, so that for each TF point (t, ω) we have a corresponding $\boldsymbol{\mathcal{W}}_*$, which satisfies

$$\boldsymbol{\mathcal{W}}_* = \arg \min_{\boldsymbol{\mathcal{W}}_i \in \boldsymbol{\mathcal{W}}} \left\| \mathcal{D}_{\boldsymbol{\Psi}}(t,\omega) - \boldsymbol{\mathcal{W}}_i \boldsymbol{\mathcal{W}}_i^{\dagger} \mathcal{D}_{\boldsymbol{\Psi}}(t,\omega) \right\|_2, \tag{5.26}$$

where $\boldsymbol{\mathcal{W}}_i^{\dagger}$ is the pseudo-inverse of $\boldsymbol{\mathcal{W}}_i$, which is defined as $\boldsymbol{\mathcal{W}}_i^{\dagger} = (\boldsymbol{\mathcal{W}}_i^{\top} \boldsymbol{\mathcal{W}}_i)^{-1} \boldsymbol{\mathcal{W}}_i^{\top}$.

For a matrix $\mathbf{W}$ of size $N \times M$ $(M > N)$, we want to derive the sub-matrices $\boldsymbol{\mathcal{W}}_i$ of size $N \times M'$, where its columns are excerpted to be independent. Thus, if M' is more than N, the columns of the sub-matrices must be non-independent. There will be exist at least one column vector that can be linearly expressed by other column vectors. Therefore, M' needs to be less than or equal to N. Similar with reference [53], we set the number of columns of sub-matrices as $N-1$, i.e. each $\boldsymbol{\mathcal{W}}_i$ composed of size $N \times (N-1)$, where $M' = N-1$ pick up from total M columns that allow us to compose an optimal sub-matrix $\boldsymbol{\mathcal{W}}_*$ from all possible combinations of the candidate set $\boldsymbol{\mathcal{W}}$, so that (5.26) is satisfied.

Thus, each source in the TF domain can be estimated by

$$\hat{\mathcal{D}}_{s_j}(t,\omega) = \begin{cases} \boldsymbol{\mathcal{W}}_*^{\dagger} \mathcal{D}_{\boldsymbol{\Psi}}(t,\omega), & \text{if } j = \lambda_i, \\ 0, & \text{otherwise}, \end{cases} \tag{5.27}$$

where λ_i is the index number of the sub-matrix that implies the non-zero element of $\tilde{\mathcal{D}}_{s_j}$ at each TF point. The source estimator $\tilde{s}_i(t)$ is then obtained by converting $\hat{\mathcal{D}}_{s_i}(t,\omega)$ to the time domain using the inverse STFT.

5.4.5 Selecting from the Extracted Components

Due to the multiple subspaces representation, the proposed method forms k extracted components. Therefore, one more thing needs to be considered that is selecting n outputs from k components as the estimator of original sources. We thus use the column-wise singular value decomposition (SVD) to form each column of the original sources $\mathbf{s}$, where the estimator forms all possible k subspaces.

The major steps of the proposed algorithm for multiple subspaces representation are summarized in Algorithm 1. In stage 1: By parameterizing such subspaces, we can map the observed signals in the feature space with the coefficient matrix from the parameter space. In stage 2: We then exploit the linear mixture in the feature space that corresponds to the nonlinear mixture in the input space. Thus, by allowing multiple sources to be presented at any point in the TF domain, we can figure out the target matrix in a sparse mixture of TF vectors. Final stage: Multiple subspaces produce k extracted components $\tilde{\mathbf{s}}$, we need to select n outputs as the estimator of the original sources $\hat{\mathbf{s}}$. Thus, the recovered sources formed from each dominant left singular vector $\mathbf{U}(:,1)$ in the column-wise SVD.

Algorithm 3 Generate Polynomials of Degree 1 by Gram-Schmidt Procedure

Input: N-dimensional observed signals $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_N(t)]^\top$.
Output: The recovered signals $\hat{\mathbf{s}}(t) = [s_1(t), s_2(t), \cdots, s_M(t)]^\top$ for $t = 1, 2, \cdots, T$.

Stage 1:
for $t = 1 : T$ **do**
    Mapping the observed signals into multiple spaces
    $\mathbf{\Psi}(\mathbf{x}(t)) = \mathbf{\Xi}^\top \mathbf{\Phi}(\mathbf{x}(t))$.
end for
Stage 2:
for $i = 1 : k$ **do**
    Transform $\mathbf{\Psi}(t)$ from the time domain into TF domain

$$\mathcal{D}_{\Psi_i}(\tau, \omega) = \sum_{t=-\infty}^{\infty} \Psi_i(t) h(t-\tau) e^{-j\omega t}.$$

end for
To remove the effect of noise, we do
$\frac{\|\mathcal{D}_{\mathbf{\Psi}}(t_p, \omega_k)\|}{\max_\omega\{\|\mathcal{D}_{\mathbf{\Psi}}(t_p, \omega)\|\}} > \epsilon$, where $\epsilon = 0.05$ in [53].
Minimizing (6.19) to derive a candidate matrix $\mathbf{W}$
$\mathcal{J}(\mathbf{c}_i, \eta) = \frac{1}{2}\|\boldsymbol{\pi}_i - \mathcal{D}_{\mathbf{\Pi}}\mathbf{c}_i\|_2^2 + \eta\|\mathbf{c}_i\|_1$,
where $\mathbf{W}$ is formed by the element of $\mathcal{D}_{\mathbf{\Pi}}$ that corresponding to the maximum value in $\mathbf{c}_i$.
The optimal sub-matrix $\boldsymbol{\mathcal{W}}$ can be derived by (5.26).
Convert the estimated source in the TF domain back to the time domain in (5.27).
Stage 3:
for $t = 1 : T$ **do**
    Apply SVD on matrix $\mathbf{F} = [\tilde{s}_1(:, t), \tilde{s}_2(:, t), \cdots, \tilde{s}_k(:, t)]$.
    The dominant left singular vector is the estimate of
    the t-th column of $\hat{s}$, i.e., $\hat{s}(:, t) \leftarrow \mathbf{U}(:, 1)$, where
    $\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.
end for

5.5 Experiments and Discussions

To evaluate the proposed algorithm, we performed the simulation on both synthetic data and real audio data over the underdetermined mixtures. First, using the synthetically generated data, the proposed algorithm is applied to show that the subspace matches the nonlinearity of mixing function in the time domain. Then the nonlinear problem can be separated in the feature space. Next, the recovered sources are tested on two kinds of environment.

5.5.1 Methods and Evaluation Metric

To evaluate the efficiency of the proposed algorithm, we perform a comparison with some developed conventional algorithms, such as the underdetermined BSS (UBSS) method based on the TF non-disjoint assumption [52], the underdetermined convolutive BSS (UCBSS) method[1] based on the subspace representation [2].

The performance of the recovered sources is evaluated by using three kinds of error measure. One is the Pearson correlation coefficient (PCC), which can evaluate the performance for each

[1] https://slsp.kaist.ac.kr/xe/index.php?mid=software

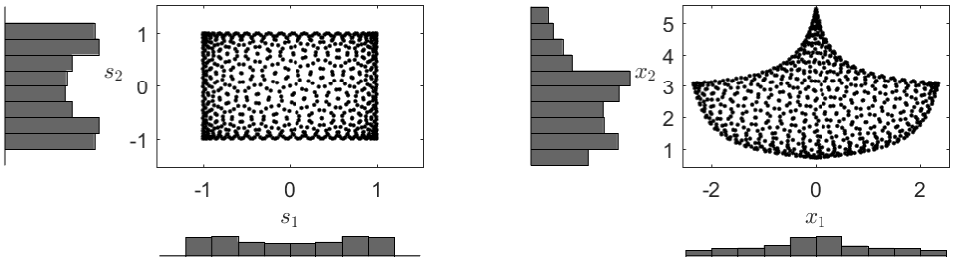


Figure 5.2: An Illustration of Nonlinear Mapping. (a) Original signals generated from two sinusoidal functions. (b) Mixture signals are modeled nonlinearly from (5.30).

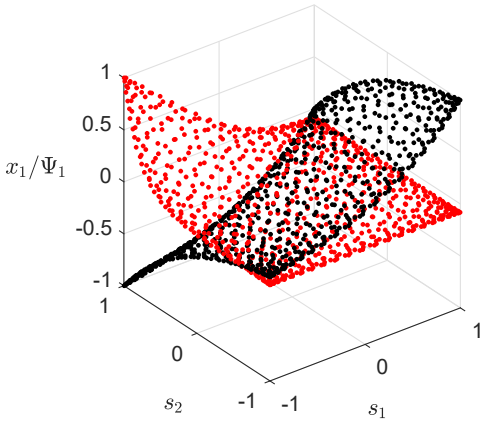


Figure 5.3: The Nonlinear Mixing $\mathbf{x}_1$ and the Subspace Constructed by Approximation Function. The black points illustrate the observed signal $\mathbf{x}_1$ in nonlinear mixing. The red points structure the subspace of best-matching. By using a coefficient matrix, the subspace can be rotated and scaled to match the nonlinear transformation.

signal on the definition of

$$\mathrm{PCC}(\mathbf{s}_i, \hat{\mathbf{s}}_i) = \frac{\mathrm{cov}(\mathbf{s}_i, \hat{\mathbf{s}}_i)}{\sigma_{\mathbf{s}_i}\sigma_{\hat{\mathbf{s}}_i}}, \tag{5.28}$$

where the recovered source and original source are denoted as $\hat{\mathbf{s}}_i$ and $\mathbf{s}_i$, respectively. $\mathrm{cov}(\cdot,\cdot)$ is the covariance between two variables and the standard deviation is denoted as σ.

The normalized mean squared error (NMSE) is another evaluation criterion used to measure the performance on the overall signals, which is defined by

$$\mathrm{NMSE}(\mathbf{s}, \hat{\mathbf{s}}) = 10\log_{10}\left(\frac{1}{M}\sum_{i=1}^{M}\min_{\delta}\frac{\|\mathbf{s}_i - \delta\hat{\mathbf{s}}_i\|_2^2}{\|\mathbf{s}_i\|_2^2}\right). \tag{5.29}$$

The scalar δ is used for controlling the scalar ambiguity.

During the separation process, the signals may be distorted especially when the sources are overlapped in their TF domain. Hence, it is necessary to measure the distortion and the artifacts introduced by the algorithm to assess the quality of separation. The BSSEVAL toolbox [158] is available online[2]. Then the source-to-distortion ratio (SDR), the source-to-interference ration

[2]http://bass-db.gforge.inria.fr/bss_eval

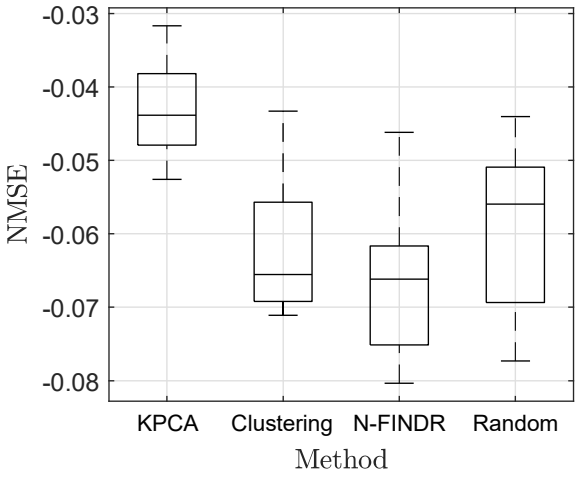


Figure 5.4: The Averaged NMSEs of Estimators Using the Different Method to Form a Set of Base.

(SIR), and the source-to-artifacts ratio (SAR) of an estimated source $\hat{s}_{ij}$ as

$$\begin{aligned}
\mathrm{SDR}_j &= 10\log_{10}\frac{\sum_{i=1}^{M}\sum_t s_{ij}(t)^2}{\sum_{i=1}^{M}\sum_t[e_{ij}^{\mathrm{spat}}(t)+e_{ij}^{\mathrm{interf}}(t)+e_{ij}^{\mathrm{artif}}(t)]^2},\\
\mathrm{SIR}_j &= 10\log_{10}\frac{\sum_{i=1}^{M}\sum_t[s_{ij}(t)^2+e_{ij}^{\mathrm{spat}}(t)]^2}{\sum_{i=1}^{M}\sum_t e_{ij}^{\mathrm{interf}}(t)^2},\\
\mathrm{SAR}_j &= 10\log_{10}\frac{\sum_{i=1}^{M}\sum_t[s_{ij}(t)+e_{ij}^{\mathrm{spat}}(t)+e_{ij}^{\mathrm{interf}}(t)]^2}{\sum_{i=1}^{M}\sum_t e_{ij}^{\mathrm{artif}}(t)^2},
\end{aligned}$$

where $\hat{s}_{ij}(t) = s_{ij}(t) + e_{ij}^{\mathrm{spat}}(t) + e_{ij}^{\mathrm{interf}}(t) + e_{ij}^{\mathrm{artif}}(t)$, s_{ij} is the target source with allowed deformation such as filtering or gain, $e_{ij}^{\mathrm{spat}}(t)$ distinct error components representing spatial distortion, $e_{ij}^{\mathrm{interf}}(t)$ accounts for the interference due to unwanted sources, and $e_{ij}^{\mathrm{artif}}(t)$ corresponds to the artifacts introduced by the separation algorithm.

5.5.2 The Effect of Multi-Subspace Representation

To see the effect of multi-subspace representation, we need to show that the subspace is extracted to approximate the varieties of nonlinearity or distortion. First, let us consider the case where the mixture signals $\mathbf{x}$ plotted in Fig. 5.2 (b) are a nonlinear mixture from two sinusoidal signals, which is also used in [44, 159] with the form of

$$\begin{aligned}
x_1(t) &= \exp(s_1(t)) - \exp(s_2(t)),\\
x_2(t) &= \exp(-s_1(t)) + \exp(-s_2(t)),
\end{aligned} \tag{5.30}$$

where $s_1(t) = \sin(0.05\pi t)$ and $s_2(t) = \sin(0.021\pi t)$ with the different frequencies. Each source has $1,000$ data points. We indicate the polynomial function of the degree 9 as a kernel function, i.e., $\mathcal{K}(\mathbf{s}_1, \mathbf{s}_2) = (\mathbf{s}_1^\top \mathbf{s}_2 + 1)^9$. Without loss of generality, we further discuss the effect of the different kernel functions. The dimensionality of subspace is set as 20.

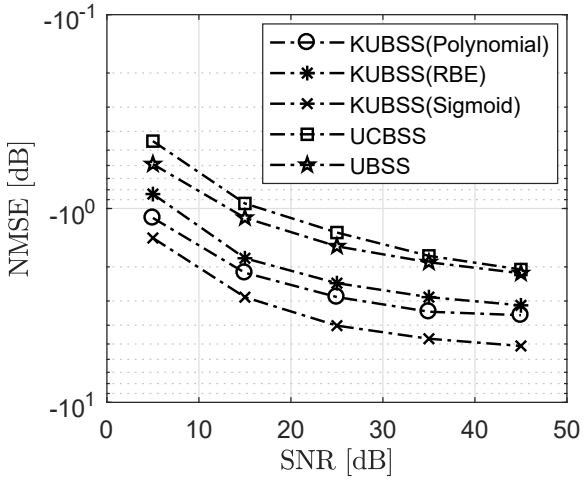


Figure 5.5: The Averaged NMSEs on the Different SNR Levels. Here the number of sources $M = 4$ and that of observations $N = 3$.

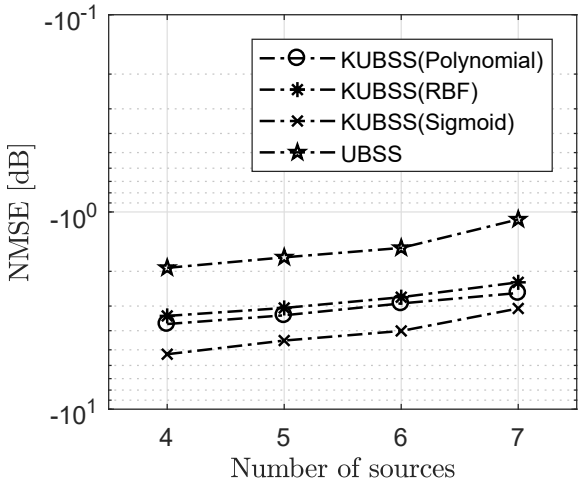


Figure 5.6: The Averaged NMSEs on the Number of Sources Increases from $M = 4$ to 7.

As shown in Fig. 5.3, the nonlinearity of the mixed signals $\mathbf{x}_1$ is comparatively strong that is plotted by black points. The observed data $\mathbf{x}_1$ is first implicitly mapped into feature space, and the effective subspace plotted by red points. Using the coefficient matrix, we can rotate and scale the subspace to match the nonlinear transformation. Relying on this effective subspace, the nonlinear problem can be linearly separable in the feature space, i.e., the original sources can be estimated linearly in the feature space by (5.20).

One of the keys in the algorithm is to find a set of orthogonal basis to study the parameterized signals in multiple subspaces. Some techniques can help that are roughly analogous in [108, 160, 161]. To perform the comparison, we employed some classical methods to extract a set of basis in the proposed algorithm, such as kernel principle component analysis (KPCA) [162], k-means [131], and random sampling. To reduce the random effect, 40 times of Monte Carlo simulations are performed.

As we can see in Fig. 5.4, using N-FINDR to extract a set of basis provides the smaller construction error in the nonlinear approximation. Either KPCA or k-means clustering is considered to obtain some vectors, which is expected to be independent. However, the method may not be

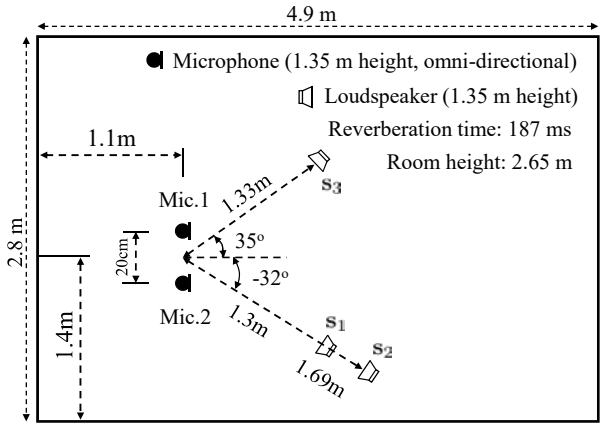


Figure 5.7: The Virtual Room Environment for Synthetic Mixtures.

Table 5.1: The Experimental Conditions.

Parameters	Values
Sampling rate	8 kHz
Number of sample points	15000 points
Window function	Hanning window
STFT frame size	1024 points (128ms)
Time frame shift	256 points (32ms)

appropriate for mixture data. This is due to the independence of the vectors, which can not be guaranteed the mutually orthogonal vectors among the basis. For further work, a set of orthonormal subspaces produced by these k vectors can represent the nonlinearity of mixing functions in the reduced data.

5.5.3 Separation of Speech and Audio Signals

To show the separation of speech and audio signals over the undertermined mixtures, the experiments are designed on two kinds of environment. Both cases use the audio data from real-world that are available in the literature [52] and online repositories[3]. The simulation is performed on the following parameter setup, where the proposed method considers the case where some examples of vector dot-product kernel. The dimensionality of subspace is set as 20. The parameter η of scalar regularization is taken as 0.001. Assume that the noise is generated from white and Gaussian with some uncorrelated data points whose variance is usually assumed to be uniform. To reduce the random effect, the simulation is repeated 20 times. The experimental conditions are summarized in Table 5.1.

The first example assumes that the mixture signals are mixed nonlinearly. The mixing functions are employed to transform $m = 4$ independent speech signals for $n = 3$ observations that are available from the literature [52], where each observation is a linear mixture of nonlinear distorted sources, i.e., $\mathbf{x}(t) = \mathbf{A}\exp(\mathbf{s}(t))$. Here, the exponential transformation provides a

[3]http://bass-db.gforge.inria.fr/BASS-dB/

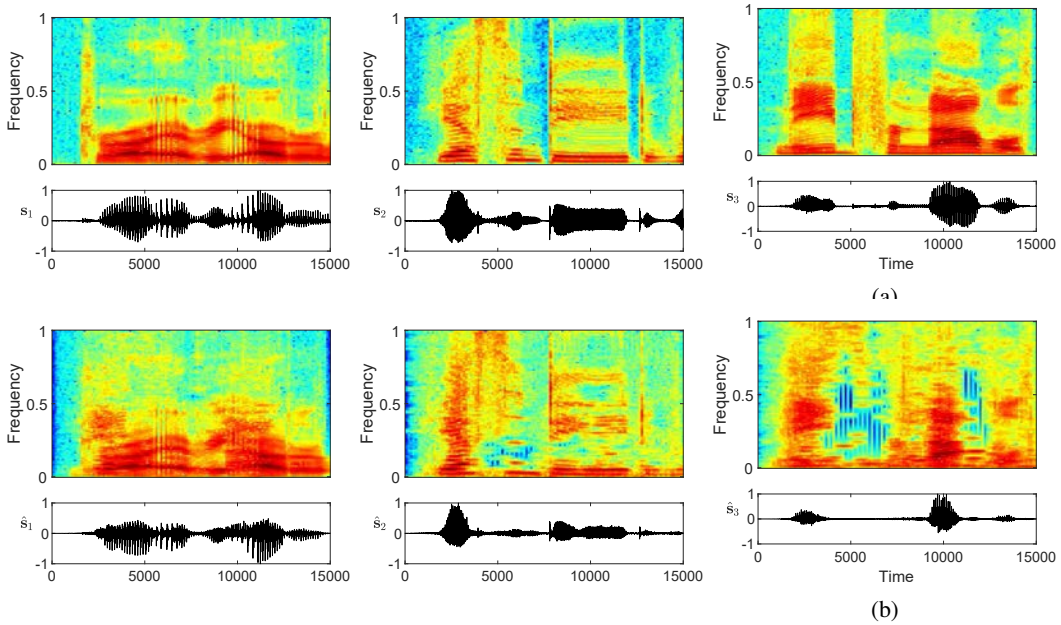


Figure 5.8: The Spectrum of Signals with Three Channels. (a) The three subfigures represent the original sources of $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$ respectively. (b) The three subfigures correspond to the recovered sources of $\hat{\mathbf{s}}_1$, $\hat{\mathbf{s}}_2$, and $\hat{\mathbf{s}}_3$, respectively.

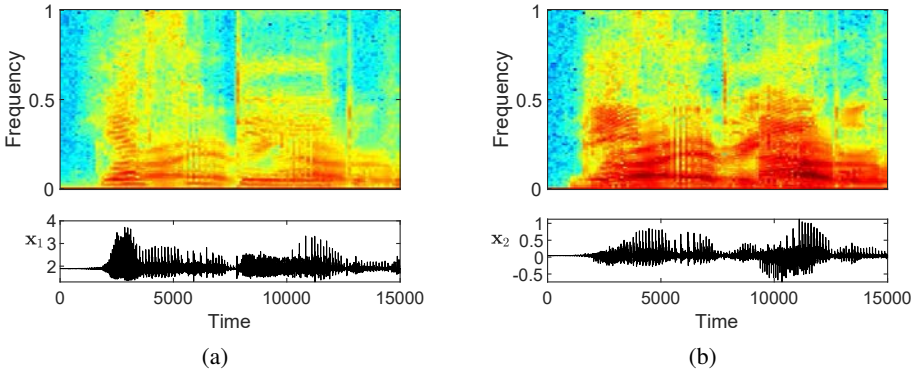


Figure 5.9: The Mixture is Achieved by Transforming 3 Original Sources to 2 Observations. The mixed signals $\mathbf{x}_1$ and $\mathbf{x}_2$ are shown in the (a) and (b), respectively.

nonlinear distortion and the matrix $\mathbf{A}$ randomly generated from a uniform distribution $U[-1, 1]$. Since there is no good path to choose a kernel function, unless we have some prior information about the data that might be helpful to determine a proper kernel function [163]. Here, we only consider the kernel function with 3 classical types, where polynomial kernel of degree 9 is given by $\mathcal{K}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^9$, Radial-basis function (RBF) of uniform variance has the definition of $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2})$, and sigmoid function is formed as $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \tanh(\mathbf{x}^\top \mathbf{y})$, respectively. The results are given under the signal-to-noise power ratio (SNR) in the range of 5 dB to 45 dB. The experiments are repeated 20 times.

In Fig. 5.5, the separation accuracy is compared with some conventional algorithms on the different SNR levels. We can see that the proposed kernel-based underdetermined blind source separation (KUBSS) algorithm consistently provides a higher accuracy over the whole SNR range. When the SNR reaches 25 dB, NMSEs decrease linearly with further increasing of SNR. Benefiting from a multi-subspace representation, the effective subspace can extract the nonlinearity or

Table 5.2: Performance Comparison of the Proposed Algorithm. The algorithm UCBSS [2] only works on the underdetermined mixture.

Active sources	Performance measure	Methods				
		KUBSS_1	KUBSS_2	KUBSS_3	UCBSS	UBSS
$\mathbf{s}_1$ and $\mathbf{s}_2$ (Collinear)	SDR	1.86	2.11	**2.37**	–	1.13
	SIR	2.24	2.49	**4.69**	–	3.75
	SAR	6.97	6.56	**6.90**	–	7.48
$\mathbf{s}_1$ and $\mathbf{s}_3$ (Non-collinear)	SDR	4.26	**4.31**	2.74	–	2.61
	SIR	4.01	**6.83**	2.72	–	2.73
	SAR	6.68	**8.59**	5.92	–	6.10
$\mathbf{s}_2$ and $\mathbf{s}_3$ (Non-collinear)	SDR	2.25	2.44	**2.77**	–	2.59
	SIR	4.09	3.87	**5.59**	–	4.02
	SAR	5.44	6.62	**7.17**	–	6.63
$\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$ (Underdetermined)	SDR	**6.97**	6.64	6.64	4.61	3.60
	SIR	4.93	**6.22**	6.21	4.49	2.47
	SAR	4.58	6.72	**6.73**	6.72	5.57

distortion caused by nonlinear mixing in kernel feature space. Moreover, this is because both UBSS and UCBSS methods are based on single source detection, which is built on the assumption that there exists only a single source or dominant energy of its corresponding single source at the TF points.

Experiment 2 shows NMSEs of the proposed algorithm where the observations are generated from the enhancement of the undetermined level, i.e., the number of sources is increased from 4 to 7 while that of observations is kept as 3. In general, a larger number of observations leads to better separation accuracy. The NMSE improvements for different combinations of sources and observations are shown in Fig. 5.6, where a set of basis is extracted using the N-FINDR approach. The kernel function also works on 3 types and 20 experiments are repeated.

Fig. 5.6 illustrates the averaged NMSEs when the number of sources increases from $M = 4$ to 7. The proposed algorithm with the "RBF" function achieved about 1.5 dB higher NMSEs against other algorithms over the whole range. In addition, 3.2 dB higher NMSEs are shown than the other algorithms when we use "Sigmoid" function. However, the performance degraded as the number of the underlying sources increased. In practice, this is due to the fact that the sources are not perfectly disjoint in the TF domain [66], which leads to the estimation error of recovered signals. As the number of sources increases, the overlap will occur in the spectra as well as the estimation error also increase.

5.5.4 Experiments Using Real Room Impulse Responses

The experiments were designed on speech data with impulse responses in an office room. The observations are collected from this room with 187 ms reverberation time. The effect of the impulse

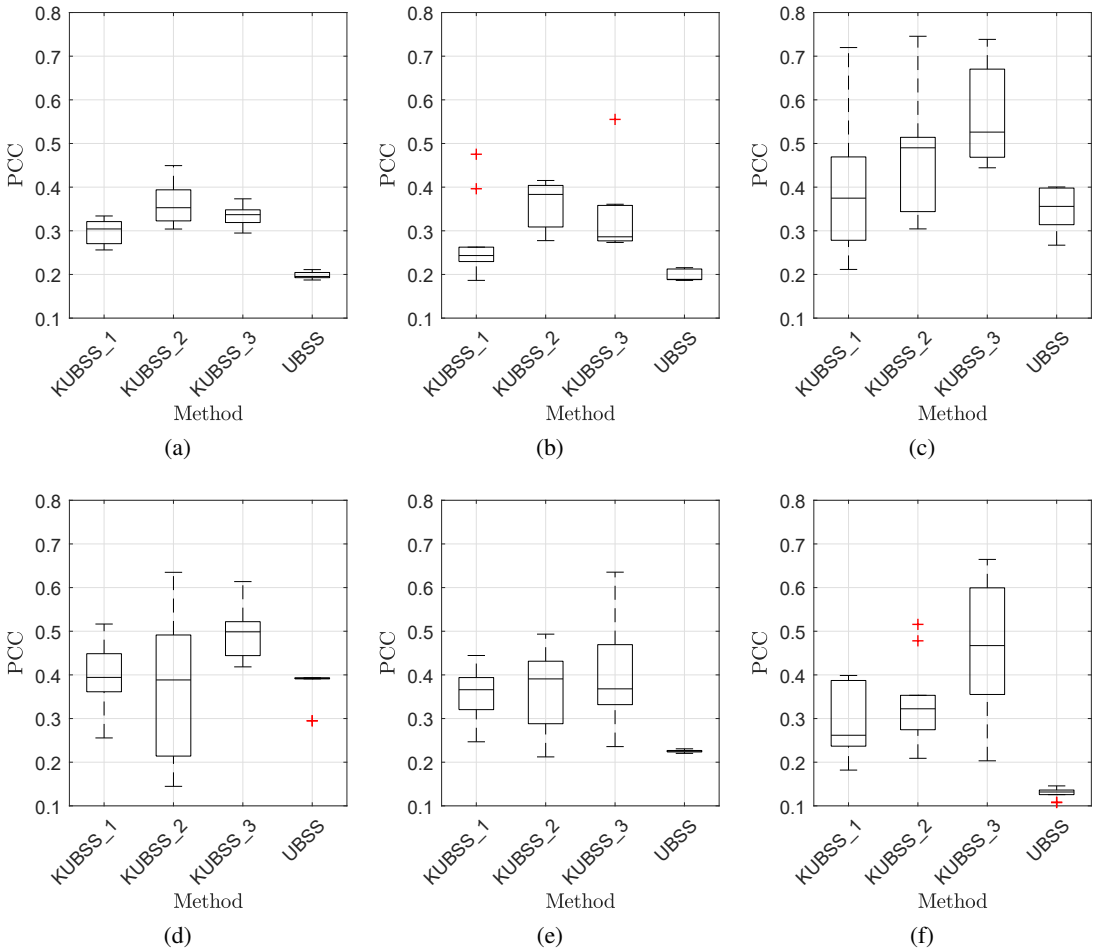


Figure 5.10: Separation of the Speech Data with Impulse Responses. The first column (a)(d) are the results from the collinear mixture of $\mathbf{s}_1$ and $\mathbf{s}_2$. The results of the non-collinear mixture are shown, respectively, in the middle column (b)(e) of $\mathbf{s}_1$ and $\mathbf{s}_3$ mixture, and the third column (c)(f) of $\mathbf{s}_2$ and $\mathbf{s}_3$ mixture. The first row (a)-(c) are PCCs of the estimated signal $\hat{\mathbf{s}}_1$. The second row (d)-(f) are PCCs of the estimated signal $\hat{\mathbf{s}}_2$.

response is measured in the face of using "Sample Champion" software that is available online[4]. Fig. 5.8 shows the original sources $\mathbf{s}(t)$ of 3 channels. Without loss of generality, the microphone, and loud speaker transfer function is neglected in the measurements [66]. The virtual room environment is illustrated in Fig. 5.7. A two-element microphone array was used for recording speech signals, which arrived in two different directions, such as $35°$ and $-32°$. It is worth noting that the source $\mathbf{s}_1$ and $\mathbf{s}_2$ are collinear that provides a challenging task using independent component analysis. The underdetermined mixture is achieved by transforming 3 original sources $\mathbf{x}(t)$ to 2 observations that are given in Fig. 5.9.

The experiments involve three scenarios, where the first case is a collinear mixture, i.e., mixed signals generated from sources $\mathbf{s}_1$ and $\mathbf{s}_2$. The second case is considered by a non-collinear mixture from $\mathbf{s}_1$ and $\mathbf{s}_3$, or $\mathbf{s}_2$ and $\mathbf{s}_3$. The third case is underdetermined mixture using all the three sources, i.e., $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$ in Fig. 5.7. Also, 3 classical kernel functions are used for comparison, such as "polynomial kernel", "RBF kernel", and "Sigmoid kernel". In the legend of the figure, they

[4]http://www.purebits.com

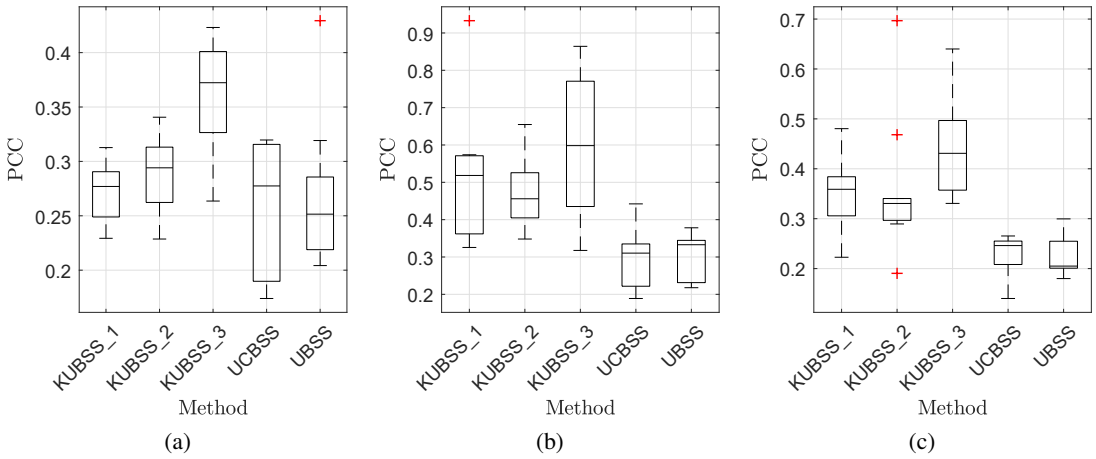


Figure 5.11: Separation of the Speech Data on the Underdetermined Mixture with the Impulse Response. (a) is the performance of estimated signal $\hat{\mathbf{s}}_1$, (b) is the performance of estimated signal $\hat{\mathbf{s}}_2$, and (c) corresponds to the estimated signal $\hat{\mathbf{s}}_3$.

are denoted as "KUBSS_1", "KUBSS_2", and "KUBSS_3", respectively, for convenience. The Pearson correlation coefficient (PCC) is used to evaluate the performance of each signal.

From Fig. 5.10, it can be seen that the algorithms can recover the original sources in all the 3 cases. We further show PCC of each channel between the original source and the recovered source using the PCC (5.28) measure. As shown in the figures, the proposed approach exhibits the promising results. This is due to the fact that the UBSS algorithm is lack of analysis of nonlinearity. In addition, the average SDR, SIR, and SAR are adopted as a performance measure of the source recovery. The performance shown in Table 5.2 are mean performances of 20 experiments. As we can see, the proposed algorithm performed better in terms of average SDR, SIR, and SAR compared with that of the UBSS methods. One can notice that the collinear mixture provides a lower accuracy than non-collinear mixture on speech sources. Therefore, a large enough angle between two sources is a crucial condition to obtain good separation performance. Some discussions have been studied in [164]. The limitation is not only for our study, but also the limitation of the separation filter obtained by ICA that forms spatial directivity [165].

Furthermore, Fig. 5.11 shows the averaged PCC on the underdetermined mixture with the impulse response. As we can see, despite adopting a similar assumption to extract sources, the proposed method exhibits a high separation accuracy compared with that of the UCBSS and UBSS methods. The main reason is that subspaces can extract the nonlinearity caused by the mixing function. As shown in Table 5.2, the proposed algorithm performs better in terms of average SDR, SIR, and SAR for situations tested. The coefficient matrix is estimated by minimizing the cost function, which is directly related to the evaluation criterion. In addition, the compared methods always require the sparsity of the sources to some extent, while the assumption may not be satisfied in reality.

5.6 Conclusions

In this chapter, we propose a multi-subspace representation based separation approach that tackles the scenario of the nonlinear and underdetermined mixture. The separation system is constructed

using the kernel methods with a multi-subspace structure. One of the keys in that algorithm is to find a set of orthogonal basis to study the parameterized signals in multiple feature spaces. We attempt to use the geometric vertices of the convex hull as the basis, which parameterizes the multi-subspace that contains the reduced vectors in the feature space. Relying on a set of an orthonormal basis, the spanned subspaces can represent the nonlinearity of mixing function in the minimum number.

Another contribution is to derive the coefficient matrix by solving an optimization problem on the coding coefficient vector. Once such subspaces are built, by allowing multiple sources to be presented at any point in the TF domain, we can figure out the target matrix in sparse mixture TF vectors with less computational cost. Finally, using this coefficient matrix, the original sources in underdetermined scenarios can be estimated. The experiments are designed on two kinds of environment, such as the signals perform nonlinear mixing, or mixing with some direction angles in a virtual room environment. The proposed approach exhibits a higher separation accuracy than that of the conventional algorithms.

Chapter 6

Polynomial Networks-based Underdetermined BSS

Chapter 5 shows a model that relies on a Kernelized multi-subspace and the sparse representation in the time-frequency (TF) domain. By generating some subspaces, the data projected into the feature space can make the nonlinear problem linearly separable. However, The mapping function do not have any optimizing property in terms of the contrast function that allows them to be ranked and evaluated. Similar to the deep architectures, Chapter 6 proposes a novel polynomial network, which extract the nonlinearity of mixing function by the network creating deeper and deeper to decrease the bias. Therefore, the approach builds the higher level representations only depend on the data, that can guarantee the robustness of structure.

Chapter 6.1 is introduction. The relative work is given in Chapter 6.2. The preliminaries included nonlinear mixture model, vanishing polynomial, and linear underdetermined BSS method are in Chapter 6.3. Chapter 6.4 introduces our proposed separation approach that construct a ϵ-vanishing polynomial networks to extract the nonlinearity. Chapter 6.5 figures out the coefficient matrix in the sparse mixture time-frequency vectors on top of network output as back propagation. Chapter 6.6 shows the experimental settings and results. Conclusions are reported in Chapter 6.7.

6.1 Introduction

Recognizing multiple signals from the multiple observations (or mixtures) received by a set of sensors is the task of source separation [166, 125]. The problem is referred to as "blind" source separation when the procedure has access only to the observations without any prior knowledge information for the mixing system. In general, most BSS algorithms assume that the number of sources is less than that of sensors, denoted as overdetermined BSS [53, 52]. However, in practice, this assumption is difficult to be satisfied since the number of sources is unknown.

Various attempts [137, 138, 159] on underdetermined BSS (UBSS) have been proposed that consider the scenario, where the number of sensors is less than that of sources. Since the mixing matrix is irreversible in this case, the recovered sources also need to be estimated even though the mixing matrix has been known. To solve this problem, a well-known framework has been proposed by exploiting the sparseness of the sources in the representation domain, such as wavelet packet transform [139] or short-time Fourier transform (STFT) [140]. For instance, the degenerate unmixing estimation technique (DUET) was proposed in [141]. The approach exploits the ratio of TF transforms of the observed signals to recover the source signals. Yilmaz et al. [142] assumed that the sources are disjoint in the TF domain. These methods work on the assumption that there exists at most one active source at any point in the TF domain. This implies that the separation performance will degrade as the number of the TF disjoints points being increased. To relax this constraint, [53, 143] proposed a scenario that allows the sources to be non-disjoint in the TF domain, however, the number of the sources that coexist at any TF point is less than that of the mixtures [53].

In the above methods, the mixing process is considered to be linear only. In fact, however, the assumption is restrictive and easy to be violated in the real-world applications [144], such as communication [145, 146], speech or audio processing [126], and biomedical engineering [148]. The problem for the nonlinear BSS is intractable solely based on the assumption that the sources are statistically independent. e.x., if x and y are two independent random variables, then $f(x)$ and

$g(y)$ are also independent for any f and g [149]. Therefore, the solutions are highly non-unique without any further constraints for the space of nonlinear mixing function [29].

Efforts on exploiting for the nonlinear approximation have involved, discovering the multi-layer architecture to capature the nonlinear structures [126, 152]. They use second-order statistic for the input components so that the vector of output are linearly independent.

Examples of polynomial neural networks include higher-order network [167], sigma-pi networks [168], and some functional link architectures [169]. The basic building block of higher-order networks is the k-th degree higher-order processing unit (HPU). However, the nonlinear behavior of the model may also cause undesired effects. For instance, the well-known phenomenon is the polynomials with high-order is used to approximate the data, which leading to unexpected ripples caused by overfitting. Consequently, typically polynomials of only second or third order are considered in practice. Such restriction to the order will leads to degradation in the extraction capability, thereby limiting the polynomial network formed by a higher order.

In this chapter, we propose a way to extend the UBSS method [52] to the nonlinear case. The derivation of our algorithm is inspired by ideas from the concept of vanishing idea [57], that gives the theorem support for existing of a finite set of polynomials. Our method attempts to construct a novel ϵ-vanishing polynomial networks (ϵ-VPNs) using vanishing component proposed in [56], but used for the different purpose. Relying on the finite vanishing polynomials, such approximated base are generated for the values attained by a set of mapping functions. Similar to the principle in deep learning, the layers of our network start with polynomials of degree 1, which has the large bias attained by this simple approximation network. To create the higher level representations of the data to decrease the bias, we next make the network deeper and deeper. Each enhancement of the degree makes the layer deeper into our network. Once the ϵ-VPNs are built that can approximate the nonlinearity or distortion caused by the mixing function. Then, we can fulfill a simple linear separation algorithm on top of this output as back propagation to adjust the network.

This work presents the advantages offered by, the deep architectures formed by a finite number of polynomials. The approach can search the number of layers automatically determined in the sense that the the error decreased as the layer being increased, ultimately almost vanish with a constant ϵ. Another contribution is to derive the coefficient matrix by solving an optimization problem on the coding coefficient vector. Once the deep architectures are built, by allowing multiple sources to be presented at any point in the TF domain, we can figure out the coefficient matrix in a sparse mixture TF vectors with less computational cost. The recovered sources thus can be derived by utilizing the coefficient matrix on top of network output as back propagation.

6.2 The Relative Work

Typically, Harmeling and Martinez [44, 151] exploit the temporal information of sources for nonlinear separation. The method performs the nonlinear BSS by mapping data into the some kernel spaces. Key assumption is that rather than indicate unique approximation, approach generates

some kernel feature spaces that are chosen enough to extract the nonlinearity of the mixing function. Finally, a selection procedure was proposed to derive the recovered sources from the extracted nonlinear components automatically.

For the observed dataset $\mathbf{x}(t) \in \mathbb{R}^n$ that are assumed to be generated by the nonlinear mixture function. To make the nonlinear problem linearly separable, the idea is to fulfill a certain condition that induces a mapping $\Phi : \mathbb{R}^n \to \mathcal{H}$ in the feature space. By using the empirical kernel map, the method defines an orthonormal basis in the form of $\boldsymbol{\Xi} := \boldsymbol{\Phi}_{\mathbf{x}}\langle\boldsymbol{\Phi}_{\mathbf{x}}, \boldsymbol{\Phi}_{\mathbf{x}}\rangle^{-\frac{1}{2}}$, where $\boldsymbol{\Phi}_{\mathbf{x}} = [\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \cdots, \Phi(\mathbf{x}_T)]$ is the mapping of data points in the feature space. By defining a basis to parameterize such subspace, the observed signals are mapped in the feature space with the coefficient matrix from a parameter space.

$$\boldsymbol{\Psi}(\mathbf{x}(t)) = \boldsymbol{\Xi}^\top \boldsymbol{\Phi}(\mathbf{x}(t)) = \langle\boldsymbol{\Phi}_{\mathbf{x}}, \boldsymbol{\Phi}_{\mathbf{x}}\rangle^{-\frac{1}{2}}\langle\boldsymbol{\Phi}_{\mathbf{x}}, \boldsymbol{\Phi}(\mathbf{x}(t)\rangle,$$

Thus, the extracted nonlinear components can be defined in the feature space as $\tilde{\mathbf{s}}(t) = \mathbf{W}^\top \boldsymbol{\Psi}(\mathbf{x}(t))$. The above equation implies that the nonlinear problem can be linearly separable in the feature space.

This method produces successful results in many experiments [42, 43]. However, a problem is that the method may fail if some sources lack specific time structures. Moreover, the method assumes the number of kernel spaces is chosen enough to approximate the nonlinearity of mixing function. Although approach construct some orthonormal base in order to extract the submanifold, the diversity of the data lead to the number of space inconsistency. In this chapter, we use vanishing ideal proposed in [56] to construct some orthonormal base. Importantly, Hilbert basis theorem [57] tells us a finite set of generates always exist that conduce to the unique solutions. Relying on the finite vanishing polynomials that generate such approximated base, we construct a ϵ-vanishing polynomial network that allowed us to solve a nonlinear problem linearly.

6.3 Model and Preliminaries

6.3.1 Nonlinear Mixture Model

The general definition of nonlinear BSS addressed in this chapter, is given as the following. Given a set of observed data $\mathcal{X} = \{\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T)\} \in \mathbb{R}^n$ that are assumed to be generated from a nonlinear, instantaneous and invertible function as

$$\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t)), \quad t = 1, \cdots, T, \tag{6.1}$$

where $\mathbf{s}(t) = [s(1), s(2), \cdots, s(T)] \in \mathbb{R}^m$ represent the original sources, and the function $\mathcal{F}$ denotes a transformation from $\mathbb{R}^m$ to $\mathbb{R}^n$. When the number of sensors is less than that of sources, i.e., $n < m$, the BSS is referred to as underdetermined. In such case, the sources need to be estimated even though the mixing matrix has been known. In addition, the solutions are highly non-unique without any further constraints for the space of nonlinear mixing function [29].

6.3.2 Vanishing Polynomial

We define a multivariate polynomial [112] that allows us to build a polynomial network so that to extract the nonlinearity or distortion caused by a mixing function. The multivariate polynomial performs a mapping $g : \mathbb{R}^n \to \mathbb{R}$ associated with $\mathbf{x} \in \mathbb{R}^n$, as the form

$$g(\mathbf{x}) = \sum_{i=0}^{\Delta} \sum_{\boldsymbol{\alpha}^{(i)}} \omega_{\boldsymbol{\alpha}^{(i)}} \prod_{j=1}^{n} x_j^{\alpha_j^{(i)}}, \tag{6.2}$$

where Δ is the degree of the polynomial, and $\boldsymbol{\alpha}^{(i)}$ ranges over all n-dimensional vectors of positive integers, such that $\sum_{j=1}^{n} \alpha_j^{(i)} = i$. $\omega \in \mathbb{R}$ is the coefficient matrix.

A concept from computer algebra to restrict polynomial is vanishing ideal [57, 56], which is defined as a set of polynomials that attend the value of zero on the dataset $\mathcal{X}$.

Definition 13 (Vanishing Ideal)**.** *Given a dataset $\mathcal{X} \subset \mathbb{R}^n$, for all $\mathbf{x} \in \mathcal{X}$, the vanishing ideal of $\mathcal{X}$ denoted as $\mathcal{I}(\mathcal{X})$, is a set of polynomials that attained the value of zero on $\mathcal{X}$ in the form of*

$$\mathcal{I}(\mathcal{X}) = \{g \in \mathcal{G}_n \mid g(\mathbf{x}) = 0 \ \textit{for} \ \forall \mathbf{x} \in \mathcal{X}\}, \tag{6.3}$$

where $\mathcal{G}_n$ is a set of polynomial of n-variates.

Since the real data are noisy that allow us to consider a tolerate value ϵ, such that the polynomials almost vanish on the data.

Definition 14 (ϵ-Vanishing Polynomial)**.** *For a tolerate value ϵ, a polynomial g is an ϵ-vanishing polynomial if $\|g(\mathbf{x})\| \leq \epsilon$ hold for $\forall \mathbf{x} \in \mathcal{X}$, where $\|\cdot\|$ denotes the Euclidean norm.*

Various attempts [56, 170] have been proposed some ways to generate such approximated vanishing components. Our approach is inspired by ideas from [112], but used for the different purpose. We attempt to construct a kind of polynomial network, where k-th layer corresponds such ϵ-vanishing polynomials of degree k. The network can search the number of layers that make deeper until the candidate dataset becomes empty.

6.3.3 Linear TF-UBSS Approach

We review a TF domain based underdetermined BSS (UBSS) method that was presented by [53] and later by defining an optimization problem on the sparse coding, we can derive the coefficient matrix. Once the polynomial network are built, we can figure out the target matrix in a sparse mixture TF vectors with less computational cost.

The discrete-time short-time Fourier transform (STFT) is given by

$$\mathcal{D}_{s_i}(\tau, \omega) = \sum_{t=-\infty}^{\infty} s_i(t) h(t - \tau) e^{-j\omega t}, \tag{6.4}$$

at frame τ and frequency bin ω, where $h(t)$ is a window function. Using STFT of (6.4), the linear BSS can be transformed into the TF domain

$$\mathcal{D}_{\mathbf{x}}(t, \omega) = \mathbf{A}\mathcal{D}_{\mathbf{s}}(t, \omega), \tag{6.5}$$

where $\mathcal{D}_{\mathbf{x}}(t,\omega) = [\mathcal{D}_{x_1}(t,\omega), \mathcal{D}_{x_2}(t,\omega), \cdots, \mathcal{D}_{x_N}(t,\omega)]^\top$ is the mixture signals in the TF domain and $\mathcal{D}_{\mathbf{s}}(t,\omega) = [\mathcal{D}_{s_1}(t,\omega), \mathcal{D}_{s_2}(t,\omega), \cdots, \mathcal{D}_{s_M}(t,\omega)]^\top$ is the STFT vector of the source signals. $\mathcal{D}_{s_i}(t,\omega)$ is the i-the source signal in the ω-the frequency bin at t time index.

To estimate the mixing vectors $\mathbf{a}_i$, the clustering algorithm is performed on the assumption in [53] that the highest densities occur around the vectors $\mathbf{a}_i$. Thus, the average values over the samples of each cluster are defined as the mixing vectors

$$\hat{\mathbf{a}}_i = \frac{1}{|C_i|} \sum_{(t,\omega)\in\Omega_i} \frac{\mathcal{D}_{\mathbf{x}}(t,\omega)}{\|\mathcal{D}_{\mathbf{x}}(t,\omega)\|}, \tag{6.6}$$

where $|C_i|$ is the number of vectors included in the same cluster.

6.4 ϵ-Vanishing Polynomial Networks-based Nonlinear Separation Approach

We will introduce a ϵ-vanishing polynomial networks (ϵ-VPNs) to estimate the original sources. The approach propose a deep structure formed by some polynomials, the layers start with polynomials of degree 1, creating the higher-level representations attend values from mapping of polynomials of higher-degree. Then, using the linear separation method, we can estimate the coefficient matrix on top of network output as back propagation.

6.4.1 The Main Idea

To make the nonlinear problem linearly separable, the idea is to generate an ϵ-VPNs, which provides a set of approximated base for the values attained by a set of recovered sources. These polynomials do not achieve the inverse of nonlinear mixing directly, but provide a good approximation for extracting the nonlinearity or distortion caused by nonlinear mixing.

Problem 1. *Given a set of data $\mathcal{X} = \{\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(T)\} \in \mathbb{R}^n$. The problem is to learn k ϵ-vanishing polynomials, such that $\{g_i(\mathbf{x}(1)), g_i(\mathbf{x}(2)), \cdots, g_i(\mathbf{x}(T))\}_{i=1}^k$ formed a set of mapping function. For the target data $\tilde{\mathbf{s}} = [\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_k]^\top$, we can find a matrix with the column vectors $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k$, so that the vector of network output*

$$\{\langle \mathbf{w}_1, \boldsymbol{\phi}_1(\mathcal{X})\rangle, \langle \mathbf{w}_2, \boldsymbol{\phi}_2(\mathcal{X})\rangle, \cdots, \langle \mathbf{w}_k, \boldsymbol{\phi}_k(\mathcal{X})\rangle : \mathbf{w}_i \in \mathbb{R}^m\}$$

are linearly independent, where $\boldsymbol{\phi}_j(\mathcal{X})$ are the projected values obtained from the mapping $g : \mathbb{R}^n \to \mathbb{R}$, and the symbol $[\cdot]^\top$ denotes the transpose operator. □

Problem 1 implies that if we generate a set of ϵ-vanishing polynomials $\{\boldsymbol{g}_1(\mathcal{X}), \boldsymbol{g}_2(\mathcal{X}), \cdots, \boldsymbol{g}_k(\mathcal{X})\}$, the outputs $\{\boldsymbol{\phi}_1(\mathcal{X}), \boldsymbol{\phi}_2(\mathcal{X}), \cdots, \boldsymbol{\phi}_k(\mathcal{X})\}$ can extract the nonlinearity or distortion caused by the nonlinear mixing function. Then a simple linear separation algorithm can be fulfilled on top of these outputs to derive the coefficient matrix $\mathbf{W}$ with the column vectors $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k$.

Similar with approach introduced in the relative work, that performs the nonlinear BSS by mapping data into the some kernel spaces on the assumption of such kernel feature spaces choosing enough to extract the nonlinearity. We utilize the same assumption, where if representation are

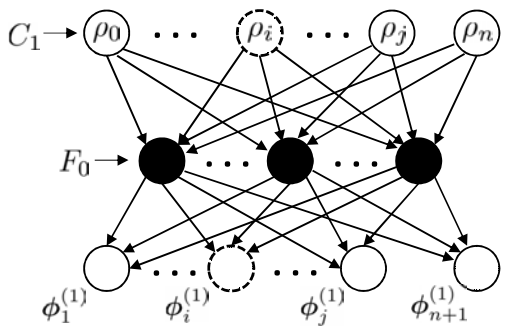


Figure 6.1: Schematic Diagram of Construction on the First-Layer. The nodes staring from the top, represent the diffusion of the zeros.

chosen enough, nonlinear probelm can be linearly separable. In this chapter, we consider the concept of vanishing ideal [57] from computer algebra that had been used for modeling a classifer in [56, 171]. Relying on the structure ϵ-VPNs with the finite vanishing polynomials, once the network are built, $\{\boldsymbol{\phi}_1(\mathcal{X}), \boldsymbol{\phi}_2(\mathcal{X}), \cdots, \boldsymbol{\phi}_k(\mathcal{X})\}$ formed a set of base can approximate original sources with the vectors $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k$. Finally, a selection procedure is performed to derive the recovered sources from the components $\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_k$ automatically.

6.4.2 Constructing the First-Layer

The polynomial of degree 1, denoted as $g^{(1)}$ is defined by a vector $\mathbf{x} = [x_1(t), x_2(t), \cdots, x_n(t)]^\top$ with the coefficient $\boldsymbol{\beta} \in \mathbb{R}^{n+1}$, such that

$$g^{(1)}(\mathbf{x}(t)) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i(t) = \langle \boldsymbol{\beta}, \boldsymbol{\rho}(\mathbf{x}(t)) \rangle, \tag{6.7}$$

where $x_i(t)$ is the i-th channel of the observations $\mathbf{x}(t)$. We use $\rho_i(\mathbf{x}(t)) = x_i(t)$ for all $i = 1, 2, \cdots, n$ for convenience, where $\rho_0(\mathbf{x}(t)) = 1$. For time t, considering all data points from $t = 1, 2, \cdots, T$, we have

$$\boldsymbol{g}^{(1)}(\mathcal{X}) = \begin{bmatrix} \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(1)) \\ \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(2)) \\ \vdots \\ \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(T)) \end{bmatrix} = \langle \boldsymbol{\beta}, \boldsymbol{\rho}(\mathcal{X}) \rangle, \tag{6.8}$$

where $\boldsymbol{\rho}_i(\mathcal{X}) = [\rho_i(\mathbf{x}(1)), \rho_i(\mathbf{x}(2)), \cdots, \rho_i(\mathbf{x}(T))]^\top$.

Thus, constructing ϵ-VPNs of 1-layer is illustrated in Fig. 6.1. The 1-layer is constructed by a set of bases that consists of all combinations searched by polynomials of degree 1. First, the input layer is from dataset $C_1 = \{\boldsymbol{\rho}_0(\mathcal{X}), \boldsymbol{\rho}_1(\mathcal{X}), \cdots, \boldsymbol{\rho}_n(\mathcal{X})\}$. We initialize two dataset, such as the set non-ϵ-vanishing polynomial (non-ϵ-VP) of degree 1, denoted as

$$F_1 = \{\boldsymbol{\eta}^{(1)}(\mathcal{X}) : \boldsymbol{\eta}^{(1)}(\mathcal{X}) = \boldsymbol{\rho}_0(\mathcal{X}) / \|\boldsymbol{\rho}_0(\mathcal{X})\|\},$$

and the sets of ϵ-vanishing polynomial (ϵ-VP) of degree 1, denoted as $V_1 = \emptyset$, respectively. Using the Gram-Schmidt algorithm, we can generate some orthogonal bases, which require

$$\gamma_i^{(1)}(\mathcal{X}) = \boldsymbol{\rho}_i(\mathcal{X}) - \sum_{\boldsymbol{\eta} \in F_0} \langle \boldsymbol{\rho}_i(\mathcal{X}), \boldsymbol{\eta}^{(1)}(\mathcal{X}) \rangle \boldsymbol{\eta}^{(1)}(\mathcal{X}), \tag{6.9}$$

less than or equal to a tolerated value $\boldsymbol{\epsilon}$, where $\gamma_i^{(1)}$ is referred as to the i-th basis in the 1-th layer.

Theorem 6. *The ϵ-VP of degree 1 denoted as $\boldsymbol{g}^{(1)}(\mathcal{X})$ vanishes on dataset $\mathcal{X}$ if and only if $\|\boldsymbol{g}^{(1)}(\mathcal{X})\| \leq \boldsymbol{\epsilon}_{T\times 1}$. It requires the vector $\boldsymbol{\beta}$ would be in the null space of the $T \times (n+1)$ matrix $\mathbf{A}_1 = [\gamma_1^{(1)}(\mathcal{X}), \gamma_1^{(2)}(\mathcal{X}), \cdots, \gamma_{n+1}^{(1)}(\mathcal{X})]$, formed as*

$$\mathbf{A}_1 \boldsymbol{\beta} = [\gamma_1^{(1)}(\mathcal{X}), \gamma_2^{(1)}(\mathcal{X}), \cdots, \gamma_{n+1}^{(1)}(\mathcal{X})]\boldsymbol{\beta} \leq \boldsymbol{\epsilon}_{T\times 1}, \tag{6.10}$$

where the tolerated value ϵ enables us to relax the effect of noise, which is a vector with the same element closed to 0. □

Up to the creation of the output layer, (6.9) can be batched by using singular value decomposition (SVD). Given a matrix $\mathbf{A}_1$ formed by $\mathbf{A}_1 = [\gamma_1^{(1)}(\mathcal{S}), \gamma_2^{(1)}(\mathcal{S}), \cdots, \gamma_{|F_1|}^{(1)}(\mathcal{X})]$, where $|F_1|$ denotes the number of elements in the set F_1. By using SVD, the matrix $\mathbf{A}_1 \in \mathbb{R}^{T\times|F_1|}$ can be decomposed as $\mathbf{A}_1 = \mathbf{L}\mathbf{D}\mathbf{U}^\top$. Using a simple matrix operation, we have

$$\mathbf{A}_1\mathbf{U} = \left[\gamma_1^{(1)}(\mathcal{X}), \gamma_2^{(1)}(\mathcal{X}), \cdots, \gamma_{|F_1|}^{(1)}(\mathcal{X})\right]\mathbf{U} = \mathbf{L}\mathbf{D}, \tag{6.11}$$

where $\mathbf{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l}_T]$ of $\boldsymbol{l}_i \in \mathbb{R}^T$. The dual representation is given by

$$\boldsymbol{g}_i^{(1)}(\mathcal{X}) = \sum_{j=1}^{|F_1|} U_{j,i}\gamma_j^{(r)}(\mathcal{X}) = \sum_{j=1}^{T} D_{j,i}\boldsymbol{l}_j = D_{i,i}\boldsymbol{l}_i, \tag{6.12}$$

where $i = 1, 2, \cdots, |F_{r-1}|$. Thus, $\boldsymbol{g}_i^{(1)}(\mathcal{X})$ is denoted as a ϵ-VP of degree 1, if and only if the diagonal element $D_{i,i}$ is less or equal to the tolerate value ϵ.

If a proper combination can be searched, which lead to $\|\boldsymbol{g}_i^{(1)}(\mathcal{X})\| \leq \boldsymbol{\epsilon}$ on the dataset $\mathcal{X}$, we update $V_1 \leftarrow V_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{X})\}$. Otherwise, $F_1 \leftarrow F_1 \bigcup \{\boldsymbol{g}_i^{(1)}(\mathcal{X})/\|\boldsymbol{g}_i^{(1)}(\mathcal{X})\|\}$ is updated. The process performs from $i = 1$ to n, at the end F_1 contains a set of non-ϵ-vanishing linear combinations which will be used for generating the 2-layer.

Therefore, the ϵ-VPNs starts with polynomials of degree 1, which have the large bias attained by this simple approximate network. To create the higher-level representations of the data to decrease the bias, we next make the network deeper and deeper. Each enhancement of the degree makes the layer deeper into our network. In particular, our network can search the number of layers that are added until the non-ϵ-vanishing set C becomes empty.

6.4.3 Constructing the Second-Layer

To exploit the polynomials of degree 2, we need to construct a candidate set of polynomials $C_2 = \{\boldsymbol{\rho}_{i,j}(\mathcal{X})\}_{i,j=1}^n$, where $\boldsymbol{\rho}_{i,j}(\mathcal{X}) = [\rho_{i,j}(\mathbf{x}(1)), \rho_{i,j}(\mathbf{x}(2)), \cdots, \rho_{i,j}(\mathbf{x}(T))]^\top$ and $\rho_{i,j}(\mathbf{x}(t)) = x_i(t)x_j(t)$

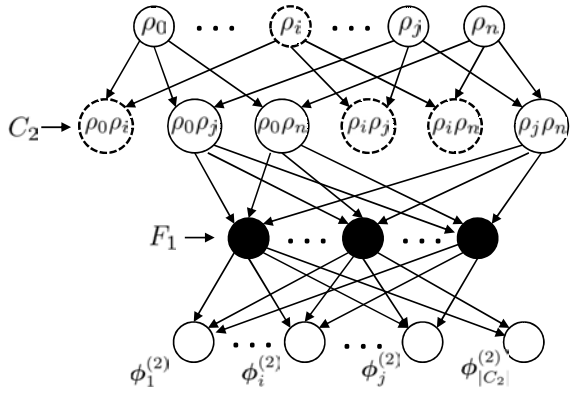


Figure 6.2: Schematic Diagram of Constructing on the Second-Layer. If the value of ρ_i is zero, then the value of all the dependent nodes are also zero. Conversely, if a basis does not attain zero, then all the nodes that make up this basis must have the values different with zero.

for all $i, j = 1, 2, \cdots, n$. Each polynomial of degree 2 takes the form

$$g^{(2)}(\mathbf{x}(t)) = \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(t)) + \sum_{i,j=1}^{n} \beta_{i,j} \rho_{i,j}(\mathbf{x}(t)). \tag{6.13}$$

By considering all the data points in $\mathcal{X}$, we have

$$\begin{aligned} \boldsymbol{g}^{(2)}(\mathcal{X}) &= \left[g^{(2)}(\mathbf{x}(1)), g^{(2)}(\mathbf{x}(2)), \cdots, g^{(2)}(\mathbf{x}(T))\right]^\top \\ &= \begin{bmatrix} \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(1)) + \sum_{i,j=1}^{n} \beta_{i,j} \rho_{i,j}(\mathbf{x}(1)) \\ \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(2)) + \sum_{i,j=1}^{n} \beta_{i,j} \rho_{i,j}(\mathbf{x}(2)) \\ \vdots \\ \sum_{i=0}^{n} \beta_i \rho_i(\mathbf{x}(T)) + \sum_{i,j=1}^{n} \beta_{i,j} \rho_{i,j}(\mathbf{x}(T)) \end{bmatrix} \\ &= \sum_{i=0}^{n} \beta_i \boldsymbol{\rho}_i(\mathcal{X}) + \sum_{i,j=1}^{n} \beta_{i,j} \boldsymbol{\rho}_{i,j}(\mathcal{X}). \end{aligned} \tag{6.14}$$

As before, we can generate the ϵ-VP of degree 2 via finding the null space of matrix $\mathbf{A}_2 = [\mathbf{A}_1, \boldsymbol{\gamma}_{1,1}^{(2)}(\mathcal{X}), \boldsymbol{\gamma}_{2,2}^{(2)}(\mathcal{X}), \cdots, \boldsymbol{\gamma}_{n,n}^{(2)}(\mathcal{X})]$. However, to generate a set of basis of degree 2, the process needs to search all $n^2 + n + 1$ combinations. Instead, using the deep architecture, we can find a sparse representation to form a set of required bases.

Fig. 6.2 illustrates the generating the 2-layer, which is constructed by some bases that consists of polynomials of degree 2. Consider a quickly running on the computational problem, we utilize a feasible pruning, i.e., sparsity of the input dataset. Assume that the dotted node ρ_i starting from the top represents the diffusion of the zeros. Namely, ρ_i generates an ϵ-VP in 1-layer, then the value of all the dependent nodes are also zero. Conversely, if a basis does not attain zero, then all the nodes that make up this basis must have the values different with zero.

Theorem 7. *Let $\boldsymbol{g}^{(2)}(\mathcal{X})$ be a set of ϵ-VPs of degree 2. It can be constructed by*

$$\boldsymbol{g}^{(2)}(\mathcal{X}) = \sum_{i_1, i_2} \boldsymbol{g}_{i_1}^{(1)} \boldsymbol{g}_{i_2}^{(1)}$$

Algorithm 4 Generating the ϵ-VP and non-ϵ-VP of degree r.

Input: C_r, F_{r-1}, $\mathcal{X}$ and tolerance ϵ.

1: **for** $i = 1$ to $|F_{r-1}|$ **do**
2: $\quad \boldsymbol{\gamma}_i^{(r)}(\mathcal{X}) = \boldsymbol{\rho}_{i_1,i_2,\cdots,i_r}(\mathcal{X}) - \sum_{\boldsymbol{\eta}^{(r-1)} \in F_{r-1}} \langle \boldsymbol{\rho}_{i_1,i_2,\cdots,i_r}(\mathcal{X}), \boldsymbol{\eta}^{(r-1)}(\mathcal{X}) \rangle \boldsymbol{\eta}^{(r-1)}(\mathcal{X})$,
3: **end for**
4: Decompose the matrix $\mathbf{A}_r = [\boldsymbol{\gamma}_1^{(r)}(\mathcal{X}), \boldsymbol{\gamma}_2^{(r)}(\mathcal{X}), \cdots, \boldsymbol{\gamma}_{|F_{r-1}|}^{(r)}(\mathcal{X})]$ using SVD, i.e., $\mathbf{A}_r = \mathbf{LDU}^\top$.
5: **for** $i = 1$ to $|F_{r-1}|$ **do**
6: $\quad \boldsymbol{g}_i^{(r)}(\mathcal{X}) = \sum_{j=1}^{|F_{r-1}|} U_{j,i} \boldsymbol{\gamma}_j^{(r)}(\mathcal{S}) = \sum_{j=1}^{T} D_{j,i} \boldsymbol{l}_j = D_{i,i} \boldsymbol{l}_i$,
7: $\quad$ **if** $D_{j,i} \leq \epsilon$ **then**
8: $\quad\quad V_r \leftarrow V_r \bigcup \{\boldsymbol{g}_i^{(r)}(\mathcal{X})\}$
9: $\quad$ **else**
10: $\quad\quad F_r \leftarrow F_r \bigcup \{\boldsymbol{g}_i^{(r)}(\mathcal{X}) / \|\boldsymbol{g}_i^{(r)}(\mathcal{X})\|\}$
11: $\quad$ **end if**
12: **end for**

Output:

1: A set of ϵ-VPs of degree r;
2: A set of non-ϵ-VPs of degree r.

$$= \sum_{j_1,j_2} \boldsymbol{\gamma}_{j_1}^{(1)} \boldsymbol{\gamma}_{j_2}^{(1)} \Big(\sum_{i_1,i_2 \leq l} U_{i_1,j_1}^{(1)} U_{i_2,j_2}^{(1)} \Big), \tag{6.15}$$

where $U_{i_1,j_1}^{(1)}$ *and* $U_{i_2,j_2}^{(1)}$ *denote the coefficients that make* $\boldsymbol{g}_{i_1}^{(1)} \boldsymbol{g}_{i_2}^{(1)} \neq \mathbf{0}_{T \times 1}$. □

Theorem 2 is proved in Appendix A. It implies that the polynomial of degree 2 can be generated from the assumption of products of two non-ϵ-VPs of degree 1. Since the nodes ρ_i for $i = 1, 2, \cdots, n$ generated the ϵ-VPs in 1-layer can be pruned in the process of constructing the 2-layer.

6.4.4 Constructing the High-Layers

To exploit the layer attained by a higher level representation, the above progress continues to generate the polynomials of higher-degree. For a polynomial of degree r, the set $C_r = \{\boldsymbol{\rho}_{i_1,i_2,\cdots,i_r}(\mathcal{X})\}_{i_1,i_2,\cdots,i_r=1}^{n}$ is formed by $\boldsymbol{\rho}_{i_1,i_2,\cdots,i_r}(\mathcal{X}) = [\rho_{i_1,i_2,\cdots,i_r}(\mathbf{x}(1)), \rho_{i_1,i_2,\cdots,i_r}(\mathbf{x}(2)), \cdots, \rho_{i_1,i_2,\cdots,i_r}(\mathbf{x}(T))]^\top$, where $\rho_{i_1,i_2,\cdots,i_r}(\mathbf{x}(t)) = x_{i_1}(t) x_{i_2}(t) \cdots x_{i_r}(t)$. To obtain the orthogonal polynomial of degree r, we have

$$\boldsymbol{\gamma}_i^{(r)}(\mathcal{X}) = \boldsymbol{\rho}_{i_1,i_2,\cdots,i_r}(\mathcal{X}) - \sum_{\boldsymbol{\eta}^{(r-1)} \in F_{r-1}} \langle \boldsymbol{\rho}_{i_1,i_2,\cdots,i_r}(\mathcal{X}), \boldsymbol{\eta}^{(r-1)}(\mathcal{X}) \rangle \boldsymbol{\eta}^{(r-1)}(\mathcal{X}),$$

where $F_{r-1} = \{\boldsymbol{\eta}_j^{(r-1)} = \frac{\boldsymbol{\rho}_j^{(r-1)}}{\|\boldsymbol{\rho}_j^{(r-1)}\|}\}$ for all $j = 1, 2, \cdots, |F_{r-1}|$, and $|F_{r-1}|$ denotes the number of elements in the set F_{r-1}.

Theorem 8. *Let $\boldsymbol{g}^{(r)}(\mathcal{X})$ be a set of ϵ-VPs of degree r. It can be constructed by*

$$\begin{aligned}\hat{\boldsymbol{g}}^{(r)}(\mathcal{X}) &= \sum_{i_1,i_2,\cdots,i_t} U_{i_1,i_2,\cdots,i_t}\boldsymbol{\gamma}_{i_1,i_2,\cdots,i_t} \\ &= \sum_{j,j_t} \boldsymbol{\gamma}_j^{(t-1)}\boldsymbol{\gamma}_{j_t}^{(1)}\Bigg(\sum_{i_t\le l} U_j^{(t-1)}U_{i_t,j_t}^{(1)}\Bigg),\end{aligned} \tag{6.16}$$

where $\boldsymbol{\gamma}_{j_t}^{(1)}$ is j_t-th non-ϵ-VP of degree 1, and $\boldsymbol{\gamma}_j^{(t-1)}$ is j-th non-ϵ-VP of degree $r-1$.

We can specify the transformation from F_{r-1} to F_r by a matrix P of size $|F_{r-1}| \times |F_1|$

$$F_r = PF_{r-1} \circ F_1$$

As we prove in Appendix B, if at any stage the subspace spanned by F_{r-1} and F_1 is the same as the subspace spanned by $\tilde{F}_{r-1}$ and F_1, then our network can span the values of all polynomials of any degree over the training data.

Constructing the Output Layer After 1 iterations (for some), we end up with a matrix F, whose columns form a basis for all values attained by polynomials of degree 1 over the training data. Moreover, each column is exactly the values attained by some node in our network over the training instances

6.5 Constructing the Output Layer

Once the deep architectures are built, by allowing multiple sources to be presented at any point in the TF domain, we can figure out the coefficient matrix in a sparse mixture TF vectors with less computational cost. The recovered sources thus can be derived by utilizing the coefficient matrix on top of network output as back propagation.

6.5.1 Coefficient Matrix Identification

Once the basis $\{\Phi_i(\mathbf{x}(1)), \Phi_i(\mathbf{x}(2)), \cdots, \Phi_i(\mathbf{x}(T))\}_{i=1}^k$ are built that can approximate the non-linearity or distortion caused by the mixing function. Then, we fulfill a simple linear separation algorithm on top on this output as back propagation to adjust the network. Here, we use a kind of UBSS method [53] on the generated basis to derive the coefficient matrix $\mathbf{W}$. Using discrete-time short-time Fourier transform (STFT), the linear relation $\tilde{\mathbf{s}} = \mathbf{W}^\dagger\boldsymbol{\Phi}$ in time-frequency (TF) domain has

$$\mathcal{D}_{\boldsymbol{\Phi}}(t,\omega) = \tilde{\mathbf{W}}\hat{\mathcal{D}}_{\mathbf{s}_i}(t,\omega), \tag{6.17}$$

where $\mathcal{D}_{\boldsymbol{\Phi}}(t,\omega) = [\mathcal{D}_{\Phi_1}(t,\omega), \mathcal{D}_{\Phi_2}(t,\omega), \cdots, \mathcal{D}_{\Phi_n}(t,\omega)]^\top$ is the projected signals in the TF domain and $\hat{\mathcal{D}}_{\mathbf{s}}(t,\omega) = [\hat{\mathcal{D}}_{s_1}(t,\omega), \hat{\mathcal{D}}_{s_2}(t,\omega), \cdots, \hat{\mathcal{D}}_{s_m}(t,\omega)]^\top$ is the STFT vector of the source signals.

Assumption 3. *Given a source signal $\mathbf{s}_i$, its STFT transformation is denoted as $\mathcal{D}_{\mathbf{s}_i}$ in the TF domain. There always exists $\mathcal{D}_{\mathbf{s}_i}$ that is dominant at all (t,ω) TF points, i.e., $|\mathcal{D}_{\mathbf{s}_i}(t,\omega)| \gg |\mathcal{D}_{\mathbf{s}_j}(t,\omega)|$ for $\forall j \neq i$.* □

The assumption implies that all sources are disjoint in the TF domain, i.e., there only one source is active on the TF point (t, ω). Then, (6.17) can be rewritten as

$$\mathcal{D}_{\mathbf{\Phi}}(t,\omega) = \hat{\mathcal{D}}_{\mathbf{s}_i}(t,\omega)\tilde{\mathbf{W}}_i, \tag{6.18}$$

where the TF feature matrix $\mathcal{D}_{\mathbf{\Phi}}(t,\omega)$ can be represented by the i-th column vector $\tilde{\mathbf{W}}_i$ with a multiplicative coefficient $\mathcal{D}_{\hat{\mathbf{s}}_i}(t,\omega)$. This implies that the target matrix $\tilde{\mathbf{W}}_i$ can be a linear combination of a few numbers of sample points from the matrix $\mathcal{D}_{\mathbf{\Phi}}(t,\omega)$ with the coefficient $\hat{\mathcal{D}}_{\mathbf{s}_i}(t,\omega)$.

We next formulate the problem of (6.18) by using a sparse direction for TF representation of the mixture TF matrix $\mathcal{D}_{\mathbf{\Phi}}(t,\omega)$. Let $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_L$ be the reshaped vector of all the mixture TF matrix $\mathcal{D}_{\mathbf{\Phi}}$, and L is the number of TF points (t,ω). We can define a one row vector $\mathbf{\Pi} \triangleq [\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_L]$ that is row-wise stacked together to be generated by the mixture TF matrix $\mathcal{D}_{\mathbf{\Phi}}$ at all (t,ω).

The further solution of (6.19) is the sparse representation of the TF feature vector $\mathcal{D}_{\mathbf{\Pi}}$, that will later construct the estimation of the coefficient matrix in the TF domain.

$$\mathcal{J}(\mathbf{c}_i, \eta) = \frac{1}{2}\|\boldsymbol{\pi}_i - \mathcal{D}_{\mathbf{\Pi}}\mathbf{c}_i\|_2^2 + \eta\|\mathbf{c}_i\|_1, \tag{6.19}$$

subject to $\mathbf{c}_{ii} = 0$, where $\eta > 0$ is a scalar parameter to balance the trade-off between the sparsity and reconstruction error. To optimize $\mathbf{c}_i$, the solution consider to use Lasso criterion and solved by the iterative soft-thresholding algorithm (ISTA) [172]. (6.19) can be rewritten as

$$\mathcal{J}(\mathbf{c}_i) = f(\mathbf{c}_i) + h(\mathbf{c}_i).$$

The problem of $\min_{\mathbf{c}_i} \mathcal{J}(\mathbf{c}_i)$, we could solve by quadratic approximation and leave $h(\mathbf{c}_i)$ alone

$$\begin{aligned}
\mathbf{c}_i(k) &= \arg\min_{\mathbf{c}_i} f(\mathbf{c}_i) + h(\mathbf{c}_i) \\
&= \arg\min_{\mathbf{c}_i} f(\mathbf{c}_i(k-1)) + \bigtriangledown f(\mathbf{c}_i(k-1))^\top(\mathbf{c}_i - \mathbf{c}_i(k-1)) + \frac{1}{2\lambda}\|\mathbf{c}_i - \mathbf{c}_i(k-1)\|_2^2 + h(\mathbf{c}_i) \\
&= \arg\min_{\mathbf{c}_i} \frac{1}{2\lambda}\|\mathbf{c}_i - (\mathbf{c}_i(k-1) - \lambda \bigtriangledown f(\mathbf{c}_i(k-1)))\|_2^2 + h(\mathbf{c}_i),
\end{aligned} \tag{6.20}$$

where λ is a step-size of gradient descent. Define the soft-thresholding operator

$$\mathrm{Soft}_\sigma(c_{i,j}) = \begin{cases} c_{i,j} - \sigma & \text{if } c_{i,j} > \sigma, \\ 0 & \text{if } -\sigma \le c_{i,j} \le \sigma, \\ c_{i,j} + \sigma & \text{if } c_{i,j} < -\sigma. \end{cases}$$

Hence, the proximal gradient [173] is used for updating

$$\mathbf{c}_i^+ = \mathrm{Soft}_{\lambda\eta}(\mathbf{c}_i + \lambda\mathcal{D}_{\mathbf{\Pi}}^\top(\boldsymbol{\pi}_i - \mathcal{D}_{\mathbf{\Pi}}\mathbf{c}_i)). \tag{6.21}$$

Once a sparse coding problem is built, the solution can be obtained by solving the convex optimization problem. Here, we use the l_1-Homotopy method in [156] to calculate the redundant dictionary $\mathbf{c}_i$ of (6.19).

6.5.2 Source Recovery

Since the mixing matrix is not irreversible in the UBSS [157], the recovered sources also need to be estimated even though the mixing matrix has been known. Therefore, we derive the sub-matrix $\hat{\mathbf{W}}$ on the following assumption.

Definition 15. *Given a matrix* $\mathbf{W}$ *of size* $n \times m$*, for any sub-matrices* $\hat{\mathbf{W}}_i$ *composed of size* $n \times (n-1)$*, there are* $\binom{m}{n-1}$ *elements included in the set of* $\hat{\mathbf{W}}$*, that is*

$$\hat{\mathbf{W}} = \{\hat{\mathbf{W}}_i | \hat{\mathbf{W}}_i = [\hat{\mathbf{W}}_{\lambda_1}, \hat{\mathbf{W}}_{\lambda_2}, \cdots, \hat{\mathbf{W}}_{\lambda_{m-1}}]\}. \tag{6.22}$$

The condition is easily met and hence not restrictive for audio data [53].

Thus, for any given mixture TF vector $\mathcal{D}_{\mathbf{\Phi}}(t,\omega)$, there must exist an optimal sub-matrix $\hat{\mathbf{W}}_* = [\hat{\mathbf{W}}_{\lambda_1}, \hat{\mathbf{W}}_{\lambda_2}, \cdots, \hat{\mathbf{W}}_{\lambda_{m-1}}]$ at each TF point (t,ω), such that

$$\hat{\mathbf{W}}_* = \arg \min_{\hat{\mathbf{W}}_i \in \hat{\mathbf{W}}} \left\| \mathcal{D}_{\mathbf{\Phi}}(t,\omega) - \hat{\mathbf{W}}_i \hat{\mathbf{W}}_i^{\dagger} \mathcal{D}_{\mathbf{\Phi}}(t,\omega) \right\|_2, \tag{6.23}$$

where $\hat{\mathbf{W}}_i^{\dagger}$ is the pseudo-inverse of $\hat{\mathbf{W}}_i$, which is defined as $\hat{\mathbf{W}}^{\dagger} = (\hat{\mathbf{W}}^{\top}\hat{\mathbf{W}})^{-1}\hat{\mathbf{W}}^{\top}$.

Thus, each source in the TF domain can be estimated by

$$\hat{\mathcal{D}}_{s_j}(t,\omega) = \begin{cases} \hat{\mathbf{W}}_*^{\dagger} \mathcal{D}_{\mathbf{\Phi}}(t,\omega), & \text{if } j = \lambda_i, \\ 0, & \text{otherwise}, \end{cases} \tag{6.24}$$

where λ_i is the index number of the optimal sub-matrix that implies the non-zero element of $\hat{\mathcal{D}}_{s_j}$ at each TF point. The source estimator $\tilde{s}_i(t)$ is then obtained by converting $\hat{\mathcal{D}}_{s_i}(t,\omega)$ to the time domain using the inverse STFT.

Due to the multiple subspaces representation, the proposed method forms k extracted components. Therefore, one more thing needs to be considered that is selecting of n outputs from k components as the estimator of original sources. We thus use the column-wise singular value decomposition (SVD) to form each column of the original sources $\mathbf{s}$, where the estimator form all possible k subspaces.

The major steps of the proposed algorithm for multiple subspaces representation are summarized in Algorithm 1. In stage 1: By parameterizing such subspaces, we can map the observed signals in the feature space with the coefficient matrix from the parameter space. In stage 2: We then exploit the linear mixture in the feature space that corresponds to the nonlinear mixture in the input space. Thus, by allowing multiple sources to be presented at any point in the TF domain, we can figure out the target matrix in a sparse mixture of TF vectors. Final stage: multiple subspaces produce k extracted components $\tilde{\mathbf{s}}$, we need to select n outputs as the estimator of the original sources $\hat{\mathbf{s}}$. Thus, the recovered sources formed from each dominant left singular vector $\mathbf{U}(:,1)$ in the column-wise SVD.

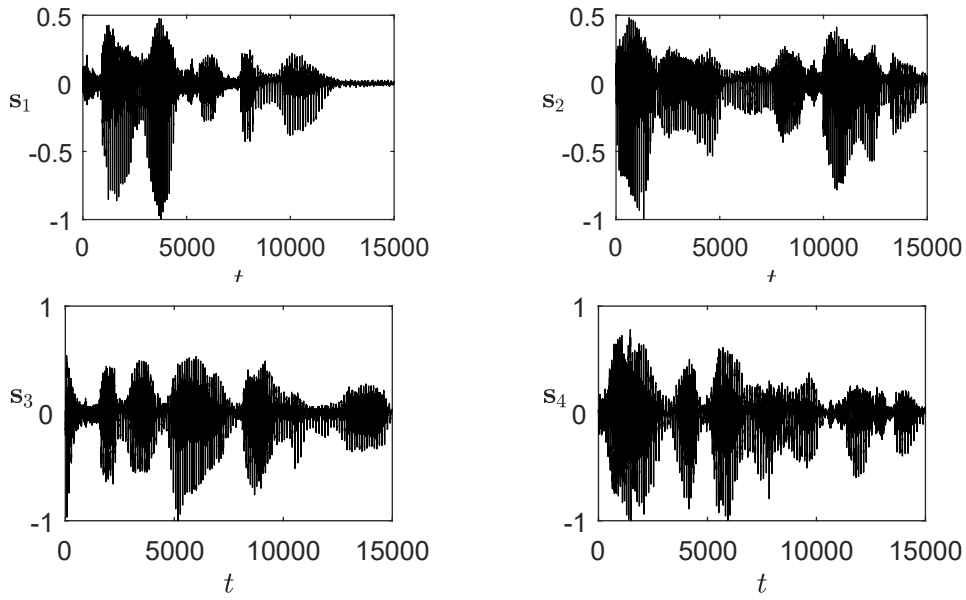


Figure 6.3: The Original Source Signals of Four Speech Signals in the Time Domain. The top two subfigures represent the original sources of $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively. The bottom subfigures correspond to the original sources of $\mathbf{s}_3$ and $\mathbf{s}_4$, respectively.

Table 6.1: The Experimental Conditions.

Parameters	Values
Sampling rate	8 kHz
Number of sample points	15000 points
Window function	Hanning window
STFT frame size	1024 points (128ms)
Time frame shift	256 points (32ms)

6.6 Experiments and Discussions

To evaluate the proposed algorithm, we performed the simulation on both synthetic data and real audio data over the underdetermined mixtures. First, using the synthetically generated data, the proposed algorithm is applied to show that the subspace matches the nonlinearity of mixing function in the time domain. Then the nonlinear problem can be separated in the feature space. Next, the recovered sources are tested on two kinds of environment.

6.6.1 Methods and Evaluation Metric

To evaluate the efficiency of the proposed algorithm, we perform a comparison with some developed conventional algorithms, such as the underdetermined BSS (UBSS) method based on the TF non-disjoint assumption [52], the underdetermined convolutive BSS (UCBSS) method[1] based on the subspace representation [2].

The performance of the recovered sources is evaluated by using two kinds of error measure. One is the Pearson correlation coefficient (PCC), which can evaluate the performance for each

[1] https://slsp.kaist.ac.kr/xe/index.php?mid=software

signal on the definition of

$$\mathrm{PCC}(\mathbf{s}_i, \hat{\mathbf{s}}_i) = \frac{\mathrm{cov}(\mathbf{s}_i, \hat{\mathbf{s}}_i)}{\sigma_{\mathbf{s}_i}\sigma_{\hat{\mathbf{s}}_i}}, \tag{6.25}$$

where the recovered source and original source are denoted as $\hat{\mathbf{s}}_i$ and $\mathbf{s}_i$, respectively. $\mathrm{cov}(\cdot,\cdot)$ is the covariance between two variables and the standard deviation is denoted as σ.

The normalized mean squared error (NMSE) is another evaluation criterion used to measure the performance on the overall signals, which is defined by

$$\mathrm{NMSE}(\mathbf{s}, \hat{\mathbf{s}}) = 10\log_{10}\left(\frac{1}{M}\sum_{i=1}^{M}\min_{\delta}\frac{\|\mathbf{s}_i - \delta\hat{\mathbf{s}}_i\|_2^2}{\|\mathbf{s}_i\|_2^2}\right). \tag{6.26}$$

The scalar δ is used for controlling the scalar ambiguity.

During the separation process, the signals may be distorted especially when the sources are overlapped in their TF domain. Hence, it is necessary to measure the distortion and the artifacts introduced by the algorithm to assess the quality of separation. The BSSEVAL toolbox [158] is available online[2]. Then the source-to-distortion ratio (SDR), the source-to-interference ration (SIR), and the source-to-artifacts ratio (SAR) of an estimated source $\hat{s}_{ij}$ as

$$\begin{aligned}
\mathrm{SDR}_j &= 10\log_{10}\frac{\sum_{i=1}^{M}\sum_t s_{ij}(t)^2}{\sum_{i=1}^{M}\sum_t[e_{ij}^{\mathrm{spat}}(t) + e_{ij}^{\mathrm{interf}}(t) + e_{ij}^{\mathrm{artif}}(t)]^2},\\
\mathrm{SIR}_j &= 10\log_{10}\frac{\sum_{i=1}^{M}\sum_t[s_{ij}(t)^2 + e_{ij}^{\mathrm{spat}}(t)]^2}{\sum_{i=1}^{M}\sum_t e_{ij}^{\mathrm{interf}}(t)^2},\\
\mathrm{SAR}_j &= 10\log_{10}\frac{\sum_{i=1}^{M}\sum_t[s_{ij}(t) + e_{ij}^{\mathrm{spat}}(t) + e_{ij}^{\mathrm{interf}}(t)]^2}{\sum_{i=1}^{M}\sum_t e_{ij}^{\mathrm{artif}}(t)^2},
\end{aligned}$$

where $\hat{s}_{ij}(t) = s_{ij}(t) + e_{ij}^{\mathrm{spat}}(t) + e_{ij}^{\mathrm{interf}}(t) + e_{ij}^{\mathrm{artif}}(t)$, s_{ij} is the target source with allowed deformation such as filtering or gain, $e_{ij}^{\mathrm{spat}}(t)$ distinct error components representing spatial distortion, $e_{ij}^{\mathrm{interf}}(t)$ accounts for the interference due to unwanted sources, and $e_{ij}^{\mathrm{artif}}(t)$ corresponds to the artifacts introduced by the separation algorithm.

6.6.2 Data and Experimental Setting

To show the separation of speech and audio signals over the undertermined mixtures, the experiments are designed on two kinds of environment. Both cases use the audio data from real-world that are available in the literature [52] and online repositories[3]. The simulation is performed on the following parameter setup, where the parameter η of scalar regularization is taken as 0.001. Assume that the noise is generated from white and Gaussian with some uncorrelated data points whose variance is usually assumed to be uniform. To reduce the random effect, the simulation is repeated 20 times. The experimental conditions are summarized in Table 6.1.

[2] http://bass-db.gforge.inria.fr/bss_eval

[3] http://bass-db.gforge.inria.fr/BASS-dB/

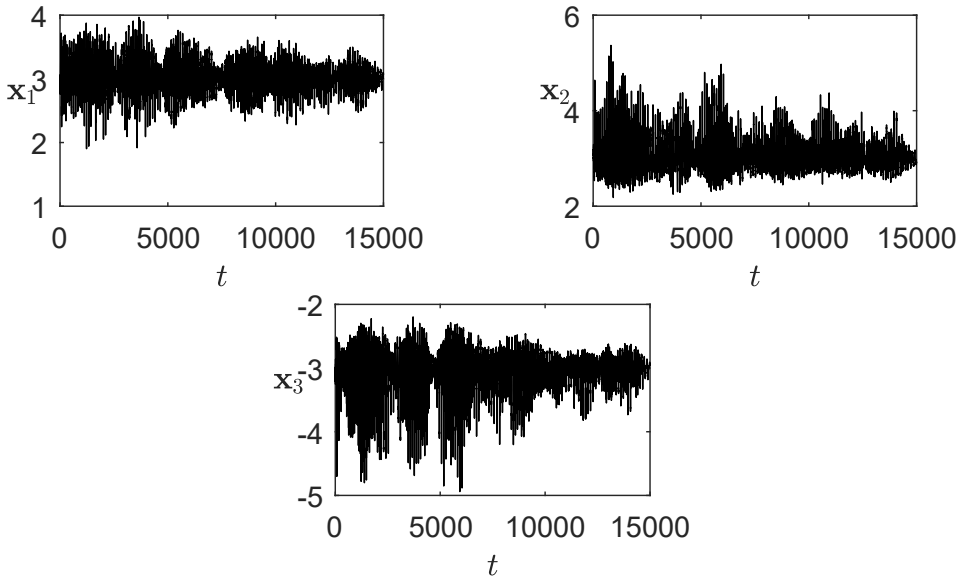


Figure 6.4: The Nonlinear Mixture is Achieved by Transforming 4 Original Sources to 3 Observations in the Time Domain. The mixed signals $\mathbf{x}_1$ and $\mathbf{x}_2$ are shown in the top two subfigures, respectively. The bottom figure corresponds to the third mixed signal $\mathbf{x}_3$.

6.6.3 Separation of Speech and Audio Signals

The first example assumes that the mixture signals are mixed nonlinearly. The mixing functions are employed to transform $m = 4$ independent speech signals for $n = 3$ observations, where each observation is a linear mixture of nonlinear distorted sources, i.e., $\mathbf{x}(t) = \mathbf{A}\exp(\mathbf{s}(t))$. Here, the exponential transformation provides a nonlinear distortion and the matrix $\mathbf{A}$ randomly generated from a uniform distribution $U[-1, 1]$. Fig. 6.3 shows the original sources $\mathbf{s}(t)$ of 4 channels in the time domain that are available from the literature [52]. To see the level of nonlinear distortion of the mixing function, the mixed signals $\mathbf{x}(t)$ of 3 channels are given in Fig. 6.4.

The second experiments, we find that with the increase of SNR, the performance of all the tested algorithms are increased. The results are given under the signal-to-noise power ratio (SNR) in the range of 5 dB to 45 dB. The experiments are repeated 20 times. From the results, the observe that the method outperforms the tested algorithms method. As both of them assume that the source signals are TF-disjoint, it means that the source signals are more sparse in quadratic TF domain than in the linear TF domain. The proposed method can still work well in such challenging situations and outperforms other methods, since the linear relations among the TF vectors at different TF points are considered.

In Fig. 6.6, the separation accuracy is compared with some conventional algorithms on the different SNR levels. We can see that the proposed vanishing polynomial networks (VPNs)-based underdetermined blind source separation algorithm consistently provides a higher accuracy over the whole SNR range. When the SNR reaches 25 dB, NMSEs decrease linearly with further increasing of SNR. Benefiting from a VPNs representation, the effective subspace can extract the nonlinearity or distortion caused by nonlinear mixing in high-dimensional space. Moreover, this is because both UBSS and UCBSS methods are based on single source detection, which is built on the assumption that there exists only a single source or dominant energy of its corresponding single source at the TF points.

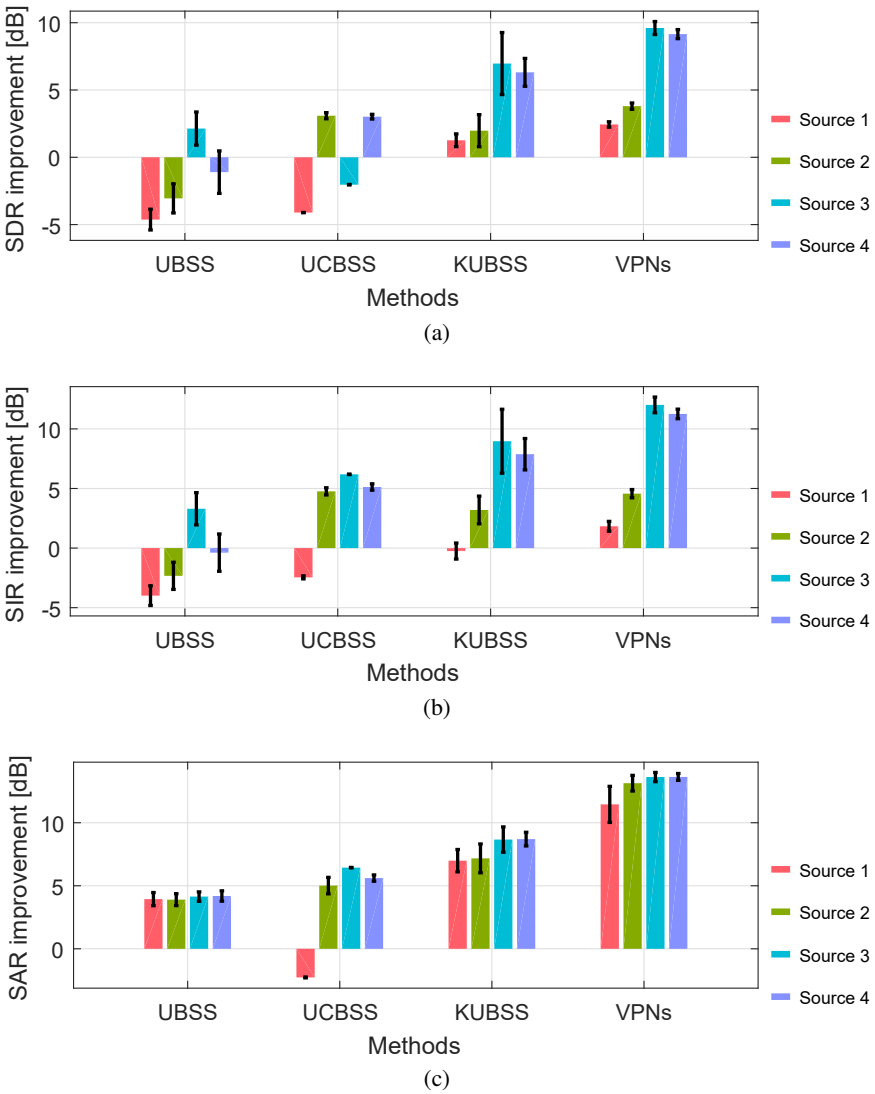


Figure 6.5: Separation of the Speech Data on the Underdetermined Mixture. (a) The average SDR improvements for speech data, (b) the average SIR improvements for speech data, and (c) the average SAR improvements for speech data.

Experiment 2 shows the averaged the source-to-distortion ratio (SDR), the source-to-interference ration (SIR), and the source-to-artifacts ratio (SAR) of the proposed algorithm where the observations are generated from the enhancement of the undetermined level, i.e., the number of sources is increased from 4 to 7 while that of observations is kept as 3. In general, a larger number of observations leads to better separation accuracy. The performance improvements for different combinations of sources and observations are shown in Fig. 6.7. 20 experiments are repeated.

Fig. 6.7 illustrates the averaged perfomance SDR, SIR, and SAR when the number of sources increases from $M = 4$ to 7. The proposed algorithm achieved about higher separation accuacy against other algorithms over the whole range. However, the performance degraded as the number of the underlying sources increased. In practice, this is due to the fact that the sources are not perfectly disjoint in the TF domain [66], which leads to the estimation error of recovered signals. As the number of sources increases, the overlap will occur in the spectra as well as the estimation error also increase.

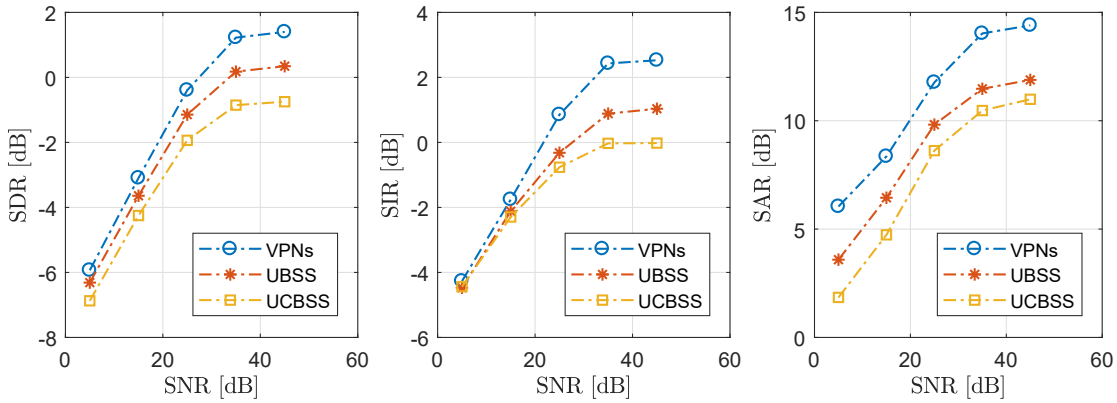


Figure 6.6: Performance Comparison of the Proposed Algorithm and the Tested Algorithms on the Different SNR Levels.

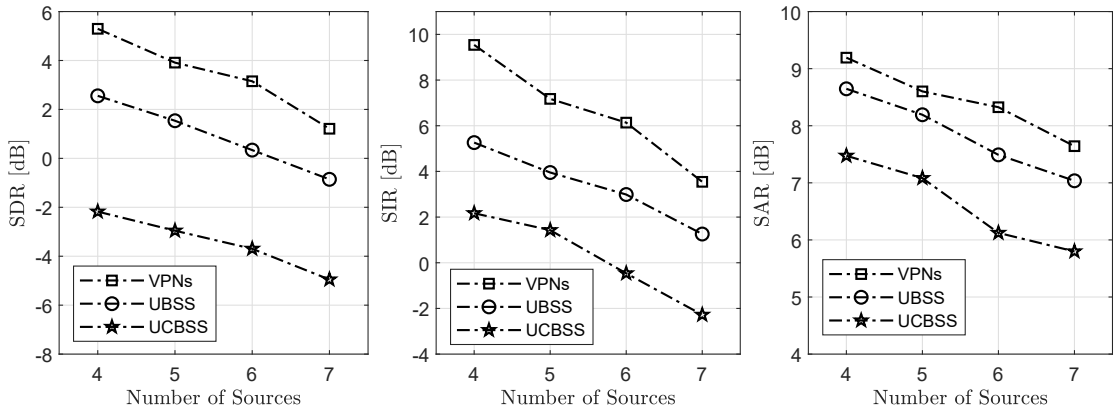


Figure 6.7: Performance Comparison of the Proposed Algorithm and the Tested Algorithms on the Number of Sources Increased from $M = 4$ to 7.

6.7 Conclusions

In this chapter, similar to the deep architecture, a novel ϵ-vanishing polynomial networks (ϵ-VPNs) is proposed to extend the linear BSS method to the nonlinear and underdetermined case. The approach attempts to construct the ϵ-VPNs using some vanishing polynomials, so as to extract the nonlinearity or distortion caused by nonlinear mixing. Relying on such approximated base are generated for the values attained by a set of mapping functions, we construct the architecture with increasing expressiveness, where the layer of our network begins with the polynomial of degree 1, up to build an output layer that can represent data with a small bias by a good approximate basis. Relying on several transformations of the input data, with higher-level representation from lower-level ones, the networks are to fulfill a mapping implicitly to the high-dimensional space. Once the ϵ-VPNs are built, we can fulfill a simple linear separation algorithm on top of this output as back propagation.

Chapter 7

Conclusions and Future Work

Most natural phenomena are inherently nonlinear, yet most statistical methods used to analyse them are linear. The methods presented in this thesis address this issue by providing nonlinear generalisations of several well-known statistical models including second-order statistic (SOS), vanishing component analysis (VCA), Kernels and feature map (KFM), and blind source separation (BSS). This chapter lists the major contributions of this dissertation. Then, some possible perspective for further research will be introduces.

7.1 Summary

In Chapter 3, We present a novel separation model that relies on the temporal structure and a novel mathematical construction with a multi-subspace architecture. The approach pre-processes the data using a flexible approximation that projects the data into a high dimensional feature space. Then, by considering the temporal decorrelation as the separation criterion, we can break a nonlinear problem down into a version of the generalized joint diagonalization problem in the feature space.

- The derivation of our algorithm is inspired by the idea of an efficient multi-subspace representation to approximate such nonlinearity. Once such a representation is built, the output is constructed by solving a convex optimization problem.
- The parameters and forms of polynomials depend solely on the input data, which guarantee the robustness of the structure. We thus address the general problem without being restricted to any specific mixture or parametric model.
- In particular, the multi-subspace representation is adaptively generated solely on the observations. As the number of spanned spaces goes up, the computational complexity grows exponentially. To overcome this obstacle, we provide a feasible way to narrow the size of the candidate polynomial set. We thus generate the current polynomial only from the non-vanishing polynomial.

The model was proposed in Chapter 3, which relies on a novel mathematical construction with multi-subspace architecture. Nevertheless, the approximation function is generated adaptively depending solely on the input data. Then the true model could be different from its empirical counterpart that is assumed to be derived by some separation algorithm with the finite sample size, which is called to be mismatched or misspecified [47]. Chapter 4 provides a theoretical analysis to model introduced in Chapter 3, which forms the closed-form expressions on the mean squared error (MSE), as well as proposing a new algebraic formalization that leads to the upper bound on the performance error. The analysis stems from the performance of a mismatched estimator that accesses the finite sample size, which is explored by two parts.

- One is to derive an iterative expression from the perspective of the expectation-maximization (EM) algorithm.
- Another one is to establish the closed-form expression for bounding the covariance matrix under both the operator norm and a class of tapering estimators.

Chapter 5 exploits the separation system is constructed using the kernel methods with a multi-subspace structure that tackles the scenario of the nonlinear and underdetermined mixture. To obtain a set of basis so as to the spanned subspace could be orthonormal in the theoretical support, we propose to use the geometric vertices of data. Then we solve a linear problem by exploiting the technique of sparse coding. The coefficient matrix is adjusted by minimizing the loss function.

- One of the keys in that algorithm is to find a set of orthogonal basis to study the parameterized signals in multiple feature spaces. We attempt to use the geometric vertices of the convex hull as the basis, which parameterizes the multisubspace that contains the reduced vectors in the feature space.

- Another contribution is to derive the coefficient matrix by solving the loss function on the coding coefficient vector. Once such subspaces are built, by allowing multiple sources to be presented at any point in the TF domain, we can figure out the target matrix in a sparse mixture TF vectors with less computational cost.

Chapter 6 proposes a way to extend the underdetermined BSS method to the nonlinear case. Similar to the principle in deep learning, the layers of our network start with polynomials of degree 1, which has the large bias attained by this simple approximation network. To create the higher level representations of the data to decrease the bias, we next make the network deeper and deeper.

Each enhancement of the degree makes the layer deeper into our network. Once the ϵ-VPNs are built that can approximate the nonlinearity or distortion caused by the mixing function.

Then, we can fulfill a simple linear separation algorithm on top of this output as back propagation to adjust the network.

- This work presents the advantages offered by, the deep architectures formed by a finite number of polynomials. The approach can search the number of layers automatically determined in the sense that the the error decreased as the layer being increased, ultimately almost vanish with a constant ϵ.

- Once the deep architectures are built, by allowing multiple sources to be presented at any point in the TF domain, we can figure out the coefficient matrix in a sparse mixture TF vectors with less computational cost. The recovered sources thus can be derived by utilizing the coefficient matrix on top of network output as back propagation.

7.2 Perspectives for Further Research

Blind separation of source signals have received wide attention in various fields such as speech enhancement [101], image recognition [102], wireless communication [103], and thus have been thoroughly studied in the signal processing community. Among, it is a non-trivial task to obtain an accurate and reliable foetal electrocardiogram (FECG) in a non-invasive fashion. Problems develop due to the facts, that the electrocardiogram (ECG) also contains a maternal electrocardiogram (MECG), which can be from one-half to one-thousandth the magnitude of the MECG [174]. Moreover, the FECG will occasionally overlap the MECG and make it normally impossible to

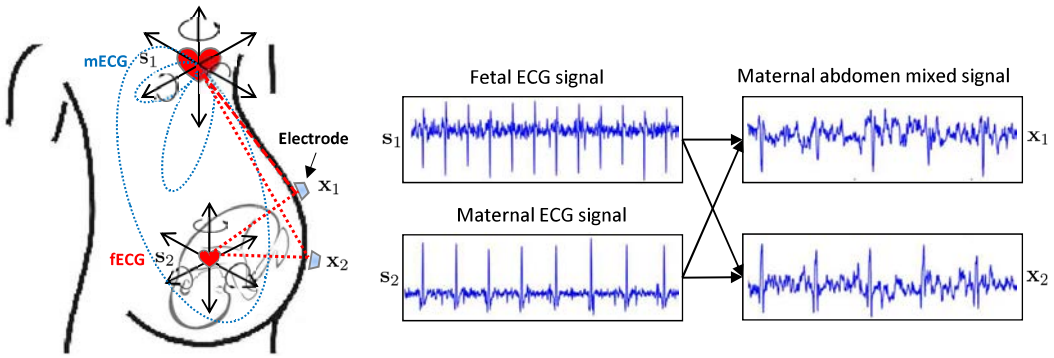


Figure 7.1: The Mixture Models of the FECG and MECG Measurements.

detect. Along with the MECG, extensive electromyographic (EMG) noise also interferes with the FECG and it can completely mask the FECG.

BSS is a very effective tool in fields where exact mathematical model is hard to generate. Therefore, one of the perspectives for my further research is extraction of FECG from Doppler electrodes located on a pregnant womans body. This can be formulated as a BSS problem, as shown in Fig. 7.1. The recordings pick up a mixture of FECG, MECG contributions, and other interferences. In the context of FECG, the purpose of BSS is to find a transformation that will separate the multivariate signal recorded on the abdomen into its additive components, which include the FECG and the MECG.

List of Author's Publications and Awards

Journals

1. L. Wang and T. Ohtsuki, "Nonlinear Blind Source Separation Unifying Vanishing Component Analysis and Temporal Structure," *IEEE Access*, vol.6, no.1, pp.42837-42850, July 2018.

2. L. Wang and T. Ohtsuki, "Underdetermined Blind Source Separation with Multi-Subspace for Nonlinear Representation," *IEEE Access*, vol.7, no.1, pp.84545-84557, June 2019.

3. L. Wang and T. Ohtsuki, "Performance Analysis for Nonlinear Separation Model with a Flexible Approximation," *Neurocomputing*. (under review)

4. Q. Ding and L. Wang, "Correlative Peak Interval Prediction and Analysis of Chaotic Sequences," *Journal of Networks*, vol.6, no.7, pp.1049-1056, 2011.

5. Q. Ding and L. Wang, "Period Extension Method and Its Implementation for a New Type of Digital Chaotic Key Sequence Generator," *Journal of Scientific Instrument*, vol.32, no.10, pp.2316-2323, 2011.

Articles on International Conferences Proceedings

1. L. Wang and T. Ohtsuki, "Underdetermined Blind Separation Using Multi-Subspace Representation in Time-Frequency Domain," *IEEE International Conference on Communication (ICC)*, Shanghai, China, May 20-24, 2019.

2. L. Wang and T. Ohtsuki, "Polynomial Networks Representation of Nonlinear Mixtures with Application in Underdetermined Blind Source Separation," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, May 12-17, 2019.

3. L. Wang and T. Ohtsuki, "Signal Restoration Based on Temporal Structure and Multi-Layer Architecture," *IEEE Global Communications Conference (Globecom)*, Abu Dhabi, UAE, Dec. 9-13, 2018.

4. L. Wang and T. Ohtsuki, "A Convergence and Asymptotic Analysis of Nonlinear Separation Model," *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Honolulu, Hawaii, Nov. 12-15, 2018.

5. X. Li, B. Qi and L. Wang, "Improved Digital Chaotic Sequence Generator Utilized in Encryption Communication," *International Conference of Intelligent Computing Technology (ICIC)*, Huangshan, China, pp.260-268, 2012.

6. X. Li, L. Wang and B. Qi, "The Improved Cycle Method and Implementation of Digital Chaotic Sequence Generator," *The International Conference on Electric Technology and Civil Engineering (ICETCE)*, Washington DC, USA, pp.3170-3173, 2012.

7. L. Wang and Q. Ding, "The Bifurcation Analysis of Digital Chaos Circuit and Its Application," *The International Workshop on Chaos-Fractals Theories and Applications (IWCFTA)*, Kunming, China, pp.242-246, 2010.

8. Q. Ding, J. Pan, L. Wang and G. Chen, "The Cipher Code Parameter Selection and Its Impact on Output Cycles," *The International Workshop on Chaos-Fractals Theories and Applications (IWCFTA)*, Shen Yang, China, pp.143-147, 2009.

9. X. Li, B. Qi and L. Wang, "A New Improved BP Neural Network Algorithm," *The Intelligent Computation Technology and Automation (ICICTA)*, Hunan, China, pp.19-22, 2009.

10. P. Zhuang, L. Wang and Q. Ding, "The Analysis and Realization of RC4 Stream Ciphers," *The International Conf. on Modeling and Simulation (ICMS)*, Manchester, UK, pp.448-452, 2009.

Articles on Domestic Conference Proceedings

1. L. Wang and T. Ohtsuki, "Source Separation Learning in ϵ-Vanishing Polynomial Network," *IEICE Society Conference*, Osaka, Japan, Sep. 10-13, 2019.

2. L. Wang and T. Ohtsuki, "Blind Source Separation in Nonlinear Mixture: Separation and a Multi-Subspace Representation," *Technical Committee on Radio Communication Systems (RCS)*, Hokkaido, Japan, Apr. 18-19, 2019.

3. L. Wang and T. Ohtsuki, "Performance Bound Approximation for Nonlinear Estimation with a Closed-Form Expression," *IEICE General Conference*, Tokyo, Japan, Mar. 19-22, 2019.

4. L. Wang and T. Ohtsuki, "A New Closed-Form Expression and Performance Analysis of Nonlinear Approximation Function," *Technical Committee on Radio Communication Systems (RCS)*, Yokosuka, Japan, Mar. 6-8, 2019.

5. L. Wang and T. Ohtsuki, "Multi-Layer Subspace Representation of Nonlinear Mixtures with Application in Nonlinear Blind Source Separation," *Multiple Innovative Kenkyu-kai Association for wireless communications (MIKA)*, Shizuoka, Japan, Sep. 26-28, 2018.

6. L. Wang and T. Ohtsuki, "Nonlinear Signal Restoration with Geometric Vertex Based Multi-subspace Mapping Approach," *IEICE Society Conference*, Kanazawa, Japan, Sep. 11-14, 2018.

7. L. Wang and T. Ohtsuki, "Performance Analysis for Nonlinear Approximation under Blind Source Separation," *IEICE General Conference*, Tokyo, Japan, Mar. 20-23, 2018.

8. L. Wang and T. Ohtsuki, "Modeling and Performance Analysis of Blind Source Separation with Nonlinear Mixing," *IEICE Technical Committee on Signal Processing (SIP)*, Tokyo, Japan, Mar. 19-20, 2018.

9. L. Wang and T. Ohtsuki, "Nonlinear Approximation Method Using for Audio Source Separation," *IEICE Technical Committee on Signal Processing (SIP)*, Takamatsu, Japan, Jan. 22-23, 2018.

10. L. Wang and T. Ohtsuki, "Vanishing Component Analysis for Nonlinear Blind Source Separation," *IEICE Society Conference*, Tokyo, Japan, Sep. 12-15, 2017.

Awards

1. 2018 IEICE RCS Active Research Award, "Blind Source Separation in Nonlinear Mixture: Separation and a Multi-Subspace Representation," Hokkaido, Japan.

2. 2011 Outstanding Master's Dissertation, Heilongjiang University.

3. 2011 The First Prize of Academic Contest Series, Heilongjiang University.

4. 2010 The First Prize of Graduate Electronic Design Competition, Chinese Institute of Electronics.

References

[1] J.-L. Roux and E. Vincent, "A Categorization of Robust Speech Processing Datasets," Cambridge, MA, USA, Tech.Rep.TR2014–116, Aug. 2014.

[2] J. Cho and C. D. Yoo, "Underdetermined Convolutive BSS: Bayes Risk Minimization Based on a Mixture of Super-Gaussian Posterior Approximation," *IEEE/ACM Trans. on Audio, Speech, and Language Process.*, vol. 23, no. 5, pp. 828–839, Mar. 2015.

[3] P. Comon, "Independent Component Analysis, A New Concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[4] T. Bell and T. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution," *Neural Comutation*, vol. 7, no. 6, pp. 1004–1034, Dec. 1995.

[5] J.-F. Cardoso and B. Laheld, "Equivariant Adaptive Source Separation," *IEEE Trans. on Signal Process.*, vol. 44, no. 12, pp. 3017–3030, Dec. 1996.

[6] A. Mansour and C. Jutten, "A Direct Solution for Blind Separation of Sources," *IEEE Trans. on Signal Process*, vol. 44, no. 3, pp. 746–748, Mar. 1996.

[7] J. Karhunen, "Neural Approaches to Independent Component Analysis and Source Separation," in *European Symposium on Artificial Neural Networks*, Apr. 1996, pp. 249–266.

[8] A. Hyvärinen and E. Oja, "A Fast Fixed Point Algorithm for Independent Component Analysis," *Neural Computation*, vol. 9, pp. 1483–1492, Jul. 1997.

[9] E. Oja, "The Nonlinear PCA Learning Rule in Independent Component Analysis," *Neurocomputing*, vol. 17, pp. 25–45, Sep. 1997.

[10] S.-I. Amari, "Natural Gradient Works Efficiently in Learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, Feb. 1998.

[11] A. Taleb and C. Jutten, "Source Separation in Post-Nonlinear Mixtures," *IEEE Trans. on Signal Process*, vol. 47, no. 10, pp. 2807–2820, Oct . 1999.

[12] T.-W. Lee, M.-S. Lewicki, M. Girolami, and T.-J. Sejnowski, "Blind Source Separation of More Sources Than Mixtures Using Overcomplete Representations," *IEEE Signal Processing Letters*, vol. 4, no. 4, pp. 87–90, Apr. 1999.

[13] D. T. Pham, "Blind Separation of Instantaneous Mixture of Sources Based on Order Statistics," *IEEE Trans. on Signal Process.*, vol. 48, no. 2, pp. 363–375, Feb. 2000.

[14] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis* . JOHN WILEY & SONS, INC, 2001.

[15] J.-F. Cardoso, "Blind Signal Separation: Statistical Principles," *Proceeding of the IEEE*, vol. 86, no. 10, pp. 2009–2025, Oct. 1998.

[16] H.-L. N. Thi and C. Juttena, "Blind Source Separation for Convolutive Mixtures," *Signal Processing*, vol. 45, no. 2, pp. 209–229, Aug. 1995.

[17] A. Mansour, C. Jutten, and P. Loubaton, "Subspace Method for Blind Separation of Sources in Convolutive Mixture," in *European Signal Processing Conference*, Sep. 1996, pp. 2081–2084.

[18] D. Yellin and E. Weinstein, "Criteria for Multichannel Signal Separation," *IEEE Trans. on Signal Process.*, vol. 42, no. 8, pp. 2158–2168, Aug. 1994.

[19] U. A. Lindgren and H. Broman, "Source Separation Using a Criterion Based on Second-Order Statistics," *IEEE Trans. on Signal Process.*, vol. 46, no. 7, pp. 1837–1850, Jul. 1998.

[20] G.-S. Fu, R. Phlypo, M. Anderson, X.-L. Li, and T. Adali, "Blind Source Separation by Entropy Rate Minimization," *IEEE Trans. on Signal Processing*, vol. 62, no. 16, pp. 4245–4255, Aug. 2014.

[21] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*. New York : Wiley, May 2000.

[22] S. Haykin, *Neural Networks and Learning Machines*. Pearson Education, 2009.

[23] A. Hyvärinen and H. Morioka, "Nonlinear ICA of Temporally Dependent Stationary Sources," in *Proc. of Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, vol. 54, 20–22 Apr. 2017, pp. 460–469.

[24] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Nonlinear Independent Components Estimation," in *arXiv:1410.8516 [cs.LG]*, Apr. 2015.

[25] G. Burel, "Blind Separation of Sources: A Nonlinear Neural Algorithm," *Neural Networks*, vol. 5, no. 6, pp. 937–947, Dec. 1992.

[26] P. Pajunen, A. Hyvärinen, and J. Karhunen., "Nonlinear Blind Source Separation by Self–organizing Maps," in *Int. Conf. on Neural Information Processing*, 1996, pp. 1207–1210.

[27] H.-H. Yang, S.-I. Amari, and A. Cichocki, "Information Theoretic Approach to Blind Separation of Sources in Nonlinear Mixture," *Signal Processing*, vol. 64, no. 3, pp. 291–300, Feb. 1998.

[28] L. B. Almeida, "ICA of Linear and Nonlinear Mixtures Based on Mutual Information," in *Inter. Joint Conf. on Neural Networks*, Jul. 2001, pp. 2991–2996.

[29] A. Hyvärinen and P. Pajunen, "Nonlinear Independent Component Analysis: Existence and Uniqueness Results," *Neural Networks*, vol. 12, no. 3, pp. 429–439, Sep. 1999.

[30] G. Marques and L. Almeida, "Separation of Nonlinear Mixtures Using Pattern Repulsion," in *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA)*, Feb. 1999, pp. 277–282.

[31] B. Ehsandoust, M. Babaie-Zadeh, B. Rivet, and C. Jutten, "Blind Source Separation in Nonlinear Mixtures: Separability and a Basic Algorithm," *IEEE Trans. on Signal Processing*, vol. 65, no. 16, pp. 4339–4352, Aug. 2017.

[32] F.-R. Bach and M.-I. Jordan, "Kernel Independent Component Analysis," *Journal of Machine Learning Researc*, vol. 3, pp. 1–48, Jul. 2002.

[33] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 1st ed. Academies Press, Feb. 2010.

[34] N. Dobigeon, J.-Y. Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero, "Nonlinear Unmixing of Hyperspectral Images: Models and Algorithms," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 82–94, Jan. 2014.

[35] M. Golbabaee, S. Arberet, and P. Vandergheynst, " Compressive Source Separation: Theory and Methods for Hyperspectral Imaging," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5096–5110, Dec. 2013.

[36] I. Meganem, P. Deliot, X. Briottet, Y. Deville, and S. Hosseini, "Physical Modelling and Nonlinear Unmixing Method for Urban Hyperspectral Images," in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, Jun. 2011, pp. 1–4.

[37] L. T. Duarte and C. Jutten, "Design of Smart Ion-Selective Electrode Arrays Based on Source Separation Through Nonlinear Independent Component Analysis," *Oil & Gas Science and TechnologyRev. IFP Energies nouvelles*, vol. 69, no. 2, pp. 239–306, Mar. 2014.

[38] F. Merrikh-Bayat, M. Babaie-Zadeh, and C. Jutten, "Linear-Quadratic Blind Source Separating Structure for Removing Show-Through in Scanned Documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 14, no. 4, pp. 319–333, Dec. 2011.

[39] G. Chollet, A. Esposito, M. Faundez-Zanuy, and M. Marinaro, *Nonlinear Speech Modeling and Applications: Advanced Lectures and Revised Selected Papers*. Springer, 2004.

[40] B. Ehsandoust, "Blind Source Separation in Nonlinear Mixtures," Ph.D. dissertation, Oct. 2018. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01885816

[41] A. Ziehe and K.-R. Müller, "TDSEP–An Effcient Algorithm for Blind Separation Using Time Structure," in *Proc. Int. Conf. on Artificial Neural Networks (ICANN)*, Nov. 1998, pp. 675–680.

[42] S. Harmeling, A. Ziehe, M. Kawanabe, B. Blankertz, and K.-R. Müller, "Nonlinear Blind Source Separation Using Kernel Feature Spaces," in *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation*, Dec. 2001, pp. 102–107.

[43] S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller, "Kernel Feature Spaces and Nonlinear Blind Source Separation," in *Advances in neural information processing systems (NIPS)*, vol. 14, Dec. 2002, pp. 761–768.

[44] S. Harmeling, A. Ziehe, and M. Kawanabe, "Kernel-based Nonlinear Blind Source Separation," *Neural Computation*, vol. 15, no. 5, pp. 1089–1124, May 2003.

[45] H. Sprekeler, T. Zito, and L. Wiskott, "An Extension of Slow Feature Analysis for Nonlinear Blind Source Separation," *Journal of Machine Learning Research*, vol. 15, pp. 921–947, 2014.

[46] Z. Wang, K. Crammer, and S. Vucetic, "Multi-Class Pegasos on a Budget ," in *Proc. of the 27th Int. Conf. on Machine Learning (ICML)*, Jun. 2010, pp. 1143–1150.

[47] S. Fortunati and F. Gini and M. S. Greco and C. D. Richmond, "Performance Bounds for Parameter Estimation under Misspecified Models," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 142–157, Nov. 2017.

[48] O. Shalvi and E. Weinstein, "Maximum Likelihood and Lower Bounds in System Identification with Non-Gaussian Inputs," *IEEE Trans. Information Theory*, vol. 40, no. 2, pp. 328–339, Mar. 1994.

[49] A. Hyvärinen and U. Köster, "FastISA: A Fast Fixed-Point Algorithm for Independent Subspace Analysis," in *Eur. Symp. Artif. Neural Netw. (ESANN)*, Apr. 2006.

[50] J.-F. Cardoso, "Multidimensional Independent Component Analysis," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 4, May 1998, pp. 1941–1944.

[51] D. Lahat, J.-F. Cardoso, and H. Messer, "Second-Order Multidimensional ICA: Performance Analysis," *IEEE Trans. on Signal Processing*, vol. 60, no. 9, pp. 4598–4610, Sep. 2012.

[52] L. Zhen, D. Peng, Z. Yi, Y. Xiang, and P. Chen, "Underdetermined Blind Source Separation Using Sparse Coding," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 28, no. 12, pp. 3102–3108, Dec. 2017.

[53] A. Aissa-El-Bey, N. Linh-Trung, K. Abed-Meraim, A. Belouchrani, and Y. Grenier, "Underdetermined Blind Separation of Nondisjoint Sources in the Time-Frequency Domain," *IEEE Trans. Signal Process.*, vol. 55, pp. 897–907, Mar. 2007.

[54] M. E. Winter, "N-Finder: An Algorithm for Fast Autonomous Spectral Endmember Determination in Hyperspectral Data," in *Image Spectrometry V, Proc. SPIE 3753*, 1999, pp. 266–277.

[55] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[56] R. Livni, D. Lehavi, S. Schein, H. Nachlieli, S. Shalev-Shwartz, and A. Globerson, "Vanishing Component Analysis," in *Proc. of Int. Conf. on Machine Learning (ICML)*, Jun. 2013, pp. 597–605.

[57] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties and Algorithm: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 3rd ed. Springer, 2007, vol. 10.

[58] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the Computational Efficiency of Training Neural Networks," *Advances in Neural Information Processing Systems*, vol. 27, pp. 855–863, Oct. 2014.

[59] L. Parra and C. Spence, "Convolutive Blind Separation of Non-Stationary Sources," *IEEE Trans. Speech Audio Processing*, vol. 8, no. 3, pp. 320–327, May 2000.

[60] K. Torkkola, "Blind Separation for Audio Signals Are We There Yet?" in *Workshop on Independent Component Analysis and Blind Signal Separation*, Jan. 1999, pp. 239–244.

[61] H.-C. Wu, J. C. Principe, and D. Xu, "Exploring the Time-Frequency Microstructure of Speech for Blind Source Separation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* , vol. 2, May 1998, pp. 1145–1148.

[62] J. M. Peterson and S. Kadambe, "A Probabilistic Approach for Blind Source Separation of Underdetermined Convolutive Mixtures," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* , vol. 6, May 2003, pp. 581–584.

[63] D. Luengo, I. Santamaria, L. Vielva, and C. Pantaleon, "Underdetermined Blind Separation of Sparse Sources with Instantaneous and Convolutive Mixtures," in *Workshop on Neural Networks for Signal Processing (NNSP)*, Sep. 2003, pp. 279–288.

[64] Y. Li, A. Cichocki, and L. Zhang, "Blind Source Estimation of FIR Channels for Binary Sources: A Grouping Decision Approach," *Signal Processing*, vol. 84, no. 12, pp. 2245–2263, Nov. 2004.

[65] H. Sawada, S. Araki, R. Mukai, and S. Makino, "Blind Extraction of A Dominant Source Signal from Mixtures of Many Sources," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* , vol. 3, 2005, pp. 61–64.

[66] V. G. Reju, S. N. Koh, and I. Y. Soon, "Underdetermined Convolutive Blind Source Separation via TimeFrequency Masking," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 1, pp. 101–116, Jan. 2010.

[67] T. Nishikawa, H. Saruwatari, and K. Shikano, "Blind source separation of acoustic signals based on multistage ica combining frequency-domain ica and time-domain ica," *IEICE Trans. Fundamentals*, 2003.

[68] X. Wang, Z. Huang, and Y. Zhou, "Underdetermined doa estimation and blind separation of non-disjoint sources in time-frequency domain based on sparse representation method," *Journal of Systems Engineering and Electronics*, 2014.

[69] Z. Koldovsky and P. Tichavsky, "Time-domain blind separation of audio sources on the basis of a complete ica decomposition of an observation space," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[70] L. Parra and C. Alvino, "Geometric source separation: Merging convolutive source separation with geometric beamforming," *IEEE Trans. Speech Audio Proc.*, 2002.

[71] Y. Luo, W. Wang, J. A. Chambers, S. Lambotharan, and I. Proudler, "Exploitation of source nonstationarity in underdetermined blind source separation with advanced clustering techniques," *IEEE Transactions on Signal Processing*, 2006.

[72] E. E. Kuruoglu, "Bayesian source separation for cosmology," *IEEE Signal Processing Magazine*, 2010.

[73] J.-T. Chien and H.-L. Hsieh, "Convex divergence ica for blind source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, 2012.

[74] S. U. N. Wood, J. Rouat, S. Dupont, and G. Pironkov, "Blind speech separation and enhancement with gcc-nmf," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[75] L. Albera, A. Ferreol, P. Chevalier, and P. Comon, "Icar: a tool for blind source separation using fourth-order statistics only," *IEEE Transactions on Signal Processing*, 2005.

[76] Z. Yang, Y. Xiang, S. Xie, S. Ding, and Y. Rong, "Nonnegative blind source separation by sparse component analysis based on determinant measure," *IEEE Transactions on Neural Networks and Learning Systems*, 2012.

[77] B. V. Gowreesunker and A. H. Tewfik, "Learning sparse representation using iterative subspace identification," *IEEE Transactions on Signal Processing*, 2010.

[78] B. Gao, W. L. Woo, and B. W. K. Ling, "Machine learning source separation using maximum a posteriori nonnegative matrix factorization," *IEEE Transactions on Cybernetics*, 2014.

[79] C. Jutten and J. Karhunen, "Advances in Blind Source Separation (BSS) and Independent Component Analysis (ICA) for Nonlinear Mixtures," *International Journal of Neural Systems*, vol. 14, no. 5, pp. 267–292, Oct. 2004.

[80] A. Taleb, "A Generic Framework for Blind Source Separation in Structured Nonlinear Models," *IEEE Trans. on Signal Processing*, vol. 50, no. 8, pp. 1819–1830, Aug. 2002.

[81] M. M. Wall and Y. Amemiya, "A Review of Nonlinear Factor Analysis Statistical Methods," Research Report RC23392, IBM, Tech. Rep., 2004.

[82] P. Werbos, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*. Van Nostrand Reinhold, New York, 1992.

[83] S. Hochreiter and J. Schmidhuber, "Feature Extraction Through LOCOCODE," *Neural Computation*, vol. 11, no. 3, pp. 679–714, Apr. 1999.

[84] S. H. J. Schmidhuber, "LOCOCODE Performs Nonlinear ICA without Knowing the Number of Sources," in *Inter. Workshop on Independent Component Analysis and Blind Signal Separation* , 1999, pp. 149–154.

[85] L. B. Almeida, "MISEP – Linear and Nonlinear ICA Based on Mutual Information," *Journal of Machine Learning Research*, vol. 4, pp. 1297–1318, Dec. 2003.

[86] ——, "Linear and Nonlinear ICA Based on Mutual Information – The MISEP Method," *Signal Processing*, vol. 84, no. 2, pp. 231–245, Feb. 2004.

[87] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[88] K. Funahashi, "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989.

[89] F. J. Theis, C. Bauer, and E. W. Lang, "Comparison of Maximum Entropy and Minimal Mutual Information in A Nonlinear Setting," *Signal Processing*, vol. 82, no. 7, pp. 971–980, Jul. 2002.

[90] B. Schölkopf and A. Smola, *Learning with Kernels*. The MIT Press, Cambridge, MA, USA, 2002.

[91] B. Schölkopf, A. Smola, and K.-R. Müller, " Nonlinear Component Analysis as A Kernel Eigenvalue Problem," *Neural Compatation*, vol. 10, no. 5, pp. 1299–1319, Dec. 1998.

[92] K. Hornik, M. Stinchcombe, and H. White, "Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks," *Neural Networks*, vol. 3, no. 5, pp. 551–560, 1990.

[93] C. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *Journal of Machine Learning Research*, vol. 7, pp. 2651–2667, Dec. 2006.

[94] V. N. Vapnik, *The Nature of Statistical Learning Theory*. SpringerVerlag New York, 1995.

[95] B. Schölkopf, A. Smola, and K.-R. Mller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, Dec. 1998.

[96] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher Discriminant Analysis with Kernels," in *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop*, 1999, pp. 41–48.

[97] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," in *14th Annual Conference on Neural Information Processing Systems (NIPS 2001)*, 2001, pp. 849–856.

[98] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Academic Press, New York, 1982.

[99] S. Julier and J. K.Uhlmann, "A General Method for Approximating Nonlinear Transformations of Probability Distributions," Technical report, Robotics, Research Group,Department of Engineering Science, University of Oxford, Tech. Rep., 1996.

[100] E. A. Wan and R. van der Merwe, *The unscented Kalman filter*. J. Wiley, New York, 2001.

[101] S. Gannot, E. Vincent, S. M.-Golan, and A. Ozerov, "A Consolidated Perspective on Multimicrophone Speech Enhancement and Source Separation," *IEEE/ACM Trans. on Audio, Speech and Language Process.*, vol. 25, no. 4, pp. 692–730, Apr. 2017.

[102] S. Amari, S.-C. Douglas, A. Cichocki, and H.-H. Yang, "Novel On-line Adaptive Learning Algorithms for Blind Deconvolution Using the Natural Gradient Approach," in *11th IFAC Symposium on Advances in Control Education*, Jun. 1997, pp. 1057–1062.

[103] W.-A. Gardner, "A New Method of Channel Identification," *IEEE Trans. on Communications*, vol. 39, no. 6, pp. 813–817, Jun. 1991.

[104] K. Zhang and L. Chan, "Minimal Nonlinear Distortion Principle for Nonlinear Independent Component Analysis," *Journal of Machine Learning Research*, vol. 9, no. 2455–2487, Nov. 2008.

[105] G. Deco and W. Brauer, "Nonlinear Higher–order Statistical Decorrelation by Volume–conserving Neural Architectures," *Neural Networks*, vol. 8, no. 525–535, Oct. 1995.

[106] Y. Wu, T. K. Doyle, and C. Fyfe, "Multi-layer Topology Preserving Mapping for K-Means Clustering," in *Proc. Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL)*, Sep. 2011, pp. 84–91.

[107] B. Qi, V. John, Z. Liu, and S. Mita, "Pedestrian Detection from Thermal Images: A Sparse Representation Based Approach," *Infrared Physics & Technology*, vol. 76, pp. 157–167, May 2016.

[108] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 22, pp. 2319–2322, Dec. 2000.

[109] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel Audio Source Separation with Deep Neural Networks," *IEEE/ACM Trans. on Audio, Speech and Language Process*, vol. 24, no. 9, pp. 1652–1664, Jun. 2016.

[110] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, "Improving Music Source Separation based on Deep Neural Networks through Data Augmentation and Network Blending," in *Inter. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, no. 261–265, Mar. 2017.

[111] E. M. Grais, G. Roma, A. J. Simpson, and M. D. Plumbley, "Discriminative Enhancement for Single Channel Audio Source Separation Using Seep Neural Networks," in *Proc. Int. Conf. on Latent Variable Analysis and Signal Separation*, no. 236–246, Feb. 2017.

[112] R. Livni, S. Shalev-Shwartz, and O. Shamir, "An Algorithm for Training Polynomial Networks," in *arXiv preprint arXiv:1304.7045*, 2013.

[113] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Trans. on Pattern Analysis and Machine Intelligence* , vol. 35, no. 8, Aug. 2013.

[114] D. Heldt, M. Kreuzer, S. Pokutta, and H. Poulisse, "Approximate Computation of Zero-dimensional Polynomial ideals," *Journal of Symbolic Computation*, vol. 44, no. 11, pp. 1566–1591, Nov. 2009.

[115] M. Donini and F. Aiolli, "Learning Deep Kernels in the Space of Dot Product Polynomials," *Machine Learning*, vol. 106, no. 9-10, pp. 1245–1269, Oct. 2017.

[116] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. of Int. Conf. on Neural iformation Processing Systems*, Dec. 2001, pp. 661–667.

[117] P. Kar and H. Karnick, "Random Feature Maps for dot Product Kernels," in *Proc. of the Int. Conf. on Artificial Intelligence and Statistic*, vol. 22, Apr. 2012, pp. 583–591.

[118] N. Pham and R. Pagh, "Fast and Scalable Polynomial Kernels via Explicit Feature Maps," in *Proc. of Int. Conf. on Knowledge Discovery and Data Mining*, Aug. 2013, pp. 239–247.

[119] H. Avron, H. Nguyen, and D. Woodruff, "Subspace Embeddings for the Polynomial Kernel," in *Proc. of Int. Conf. on Neural Information Processing Systems*, Dec. 2014, pp. 2258–2266.

[120] M. Blondel, M. Ishihata, A. Fujino, and N. Ueda, "Polynomial Networks and Factorization Machines: New Insights and Efficient Training Algorithms," in *Proc. of the Int. Conf. on Machine Learning*, vol. 48, Jun. 2016, pp. 850–858.

[121] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A Blind Source Separation Technique Using Second-Order Statistics," *IEEE Trans. on Signal Processing*, vol. 45, no. 2, pp. 434–444, Feb. 1997.

[122] J.-F. Cardoso and A. Souloumiac, "Jacobi Angles for Simultaneous Diagonalization," *Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 161–164, Jan 1996.

[123] J.-F. Cardoso, "High–Order Contrasts for Independent Component Analysis," *Neural Computation*, vol. 11, pp. 157–192, 1999.

[124] M. Babaie-Zadeh, "On Blind Source Separation in Convolutive and Nonlinear Mixtures," Ph.D. dissertation, Sep. 2002. [Online]. Available: http://sharif.edu/~mbzadeh/Publications/PublicationFiles/PhD_Thesis/2002/MBzadehPhDthesisEnglish.pdf

[125] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. John Wiley & Sons, LTD, 2002.

[126] L. Wang and T. Ohtsuki, "Nonlinear Blind Source Separation Unifying Vanishing Component Analysis and Temporal Structure," *IEEE Access*, vol. 6, pp. 42 837–42 850, July 2018.

[127] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[128] T. T. Cai and H. H. Zhou, "Optimal Rates of Convergence for Sparse Covariance Matrix Estimation," *The Annals of Statistics*, vol. 40, no. 5, pp. 2389–2420, Mar. 2012.

[129] M. Tumminello, F. Lillo, and R. N. Mantegna, "Kullback-Leibler Distance as a Measure of the Information Filtered from Multivariate Data," *Physical Review E* , 2007.

[130] M. Anderson, T. Adali, and X.-L. Li, "Joint Blind Source Separation With Multivariate Gaussian Model: Algorithms and Performance Analysis," *IEEE Trans. on Signal Processing*, vol. 60, no. 4, pp. 1672–1683, Dec. 2012.

[131] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[132] P. J. Bickel and E. Levina, "Regularized Estimation of Large Covariance Matrices," *The Annals of Statistics*, vol. 36, no. 1, pp. 199–227, Feb. 2008.

[133] T. Wei, "A Convergence and Asymptotic Analysis of the Generalized Symmetric FastICA Algorithm," *IEEE Trans. on Signal Processing* , vol. 63, no. 24, pp. 6445–6458, Aug. 2015.

[134] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint Embedding Learning and Sparse Regression: A Framework for Unsupervised Feature Selection," *IEEE Trans. on Cybernetics*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[135] D. Lahat and C. Jutten, "Joint Independent Subspace Analysis Using Second-Order Statistics," *IEEE Trans. on Signal Processing*, vol. 64, no. 18, pp. 4891–4904, Sep. 2016.

[136] K. B. Petersen and M. S. Pedersen. (2012, Nov.) The Matrix Cookbook. [Online]. Available: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3274

[137] D. Peng and Y. Xiang, "Underdetermined Blind Source Separation Based on Relaxed Sparsity Condition of Sources," *IEEE Trans. on Signal Processing*, vol. 57, no. 2, pp. 809–814, 2009.

[138] L. Zou, X. Chen, X. Ji, and Z. J. Wang, "Underdetermined Joint Blind Source Separation of Multiple Datasets," *IEEE Access*, vol. 5, pp. 7474–7487, Apr. 2017.

[139] P. Georgiev, F. Theis, and A. Cichocki, "Sparse Component Analysis and Blind Source Separation of Underdetermined Mixtures," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 992–996, Jul. 2005.

[140] A. Belouchrani, M. G. Amin, N. Thirion-Moreau, and Y. D. Zhang, "Source Separation and Localization Using Time-Frequency Distributions: An Overview," *IEEE Signal Process. Magazine*, vol. 30, no. 6, pp. 97–107, Nov. 2013.

[141] A. Jourjine, S. Rickard, and O. Yilmaz, "Blind Separation of Disjoint Orthogonal Signals: Demixing N Sources from 2 Mixtures," in *IEEE Inter. Conf. on Acoustics, Speech, and Signal Process.*, 2000.

[142] O. Yilmaz and S. Rickard, "Blind Separation of Speech Mixtures via Time-Frequency Masking," *IEEE Trans. on Signal Process.*, vol. 52, no. 7, pp. 1830–1847, Jul. 2004.

[143] R. Vidal, "Subspace clustering," *IEEE Signal Process. Magazine*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[144] C. Jutten and J. Karhunen, "Advances in Nonlinear Blind Source Separation," in *Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA)*, Apr. 2003, pp. 245–256.

[145] L. Yang, Y. Xiang, and D. Peng, "Precoding-Based Blind Separation of MIMO FIR Mixtures," *IEEE Access*, vol. 5, pp. 12 417–12 427, Jul. 2017.

[146] R. Liu, X. Zhu, Y. Jiang, X. Dong, and F. Zheng, "Blind PAPR Reduction and ICA Based Equalization for mmWave FBMC-OQAM Systems," in *IEEE International Conference on Communications (ICC)*, May 2018.

[147] L. Wang and T. Ohtsuki, "Polynomial Networks Representation of Nonlinear Mixtures with Application in Underdetermined Blind Source Separation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* , May 2019.

[148] M. Banuelos, S. Sindi, and R. F. Marcia, "Negative Binomial Optimization for Biomedical Structural Variant Signal Reconstruction," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 906–910.

[149] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1991. [Online]. Available: https://dcetit.files.wordpress.com/2013/03/papoulis-probability-random-variables-and-stochastic-processes.pdf

[150] Y. Tan, J. Wang, and J. M. Zurada, "Nonlinear Blind Source Separation Using a Radial Basis Function Network," *IEEE Trans. on Neural Networks*, vol. 12, no. 1, pp. 124–134, Jan. 2001.

[151] D. Martinez and A. Bray, "Nonlinear Blind Source Separation Using Kernels," *IEEE Trans. on Neural Networks*, vol. 14, no. 1, pp. 228–235, Jan. 2003.

[152] L. Wang and T. Ohtsuki, "Signal Restoration Based on Temporal Structure and Multi-Layer Architecture," in *IEEE Global Communications Conference (GLOBECOM2018)*, Dec. 2018.

[153] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, UK, Seventh printing with corrections, 2009.

[154] A. Ambikapathi, T.-H. Chan, W.-K. Ma, and C.-Y. Chi, "Chance-Constrained Robust Minimum-Volume Enclosing Simplex Algorithm for Hyperspectral Unmixing," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4149–4209, 2011.

[155] C.-H. Lin, C.-Y. Chi, Y.-H. Wang, and T.-H. Chan, "A Fast Hyperplane-Based Minimum-Volume Enclosing Simplex Algorithm for Blind Hyperspectral Unmixing," *IEEE Trans. on Signal Processing*, vol. 64, no. 8, pp. 1946–1961, Apr. 2015.

[156] M. S. Asif and J. Romberg, "Sparse Recovery of Streaming Signals Using l_1-Homotopy," *IEEE Trans. on Signal Process.*, vol. 62, no. 16, pp. 4209–4223, Aug. 2014.

[157] D. Peng and Y. Xiang, "Underdetermined Blind Separation of Non-Sparse Sources Using Spatial Time-Frequency Distributions," *Digital Signal Process.*, vol. 20, no. 2, pp. 581–596, Mar. 2010.

[158] E. Vincent, S. Araki, F. J. Theis, G. Nolte, and P. Bofill, "The Signal Separation Evaluation Campaign (2007-2010): Achievements and Remaining Challenges. ," *Signal Processing*, no. 92, pp. 1928–1936, 2012.

[159] L. Wang and T. Ohtsuki, "Underdetermined Blind Separation Using Multi-Subspace Representation in Time-Frequency Domain," in *IEEE International Conference on Communications (ICC2019)*, May 2019.

[160] Z. Zhang and H. Zha, "Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 313–338, Dec. 2004.

[161] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, Jan 2007.

[162] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel Principal Components Analysis," in *Inter. Conf. on Artificial Neural Networks*, Jun. 1997, pp. 583–588.

[163] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[164] V. G. Reju, S. N. Koh, and I. Y. Soon, "Partial Separation Method for Solving Permutation Problem in Frequency Domain Blind Source Separation of Speech Signals," *Neurocomputing*, vol. 71, no. 10–12, pp. 2098–2112, Jun. 2008.

[165] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Frequency-Domain Blind Source Separation of Many Speech Signals Using Near-Field and Far-Field Models," *EURASIP Journal on Applied Signal Processing*, pp. 1–13, Dec. 2006.

[166] A. Hyvärinen, "Blind Source Separation by Nonstationarity of Variance: A Cumulant based Approach," *IEEE Trans. on Neural Networks*, vol. 12, no. 6, pp. 1471–1474, 2001.

[167] C. L. Giles and T. Maxwell, "Learning, Invariance, and Generalization in a High-Order Neural Network," *Applied Optics*, vol. 26, no. 23, pp. 4972–4978, Dec. 1987.

[168] C.-K. Li, "A Sigma-Pi-Sigma Neural Network (SPSNN)," *Neural Processing Letters*, vol. 17, no. 1, pp. 1–19, Feb. 2003.

[169] Y.-H. Pao and Y. Takefuji, "Functional-Link Net Computing: Theory, System Architecture, and Functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, May 1992.

[170] R. Iraji and H. Chitsaz, "Principal Variety Analysis," in *Proc. of Conf. on Robot Learning (CoPL)*, 2017, pp. 97–108.

[171] H. Kera and Y. Hasegawa, "Spurious Vanishing Problem in Approximate Vanishing Ideal," in *arXiv preprint: arXiv:1901.08798v1*, 2019.

[172] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[173] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, "Smoothing Proximal Gradient Method for General Structured Sparse Learning," *The Annals of Applied Statistics*, vol. 6, no. 2, pp. 719–752, Feb. 2012.

[174] S. Noorzadeh, "Extraction of Fetal ECG and Its Characteristics Using Multi-Modality," Feb. 2016. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01279019